

Lab 01

How to Survive & Introduction to Git

Web Programming
DataLab, CS, NTHU

Notice

- These slides will focus on how to submit you code by using Git command line
- You can also use other Git GUI tool or built-in Git tool in other IDE/editor

Outline

- General Rule
- Introduction to Git
 - Version control
 - Git Basics
 - Try Git!
 - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

Outline

- General Rule
- Introduction to Git
 - Version control
 - Git Basics
 - Try Git!
 - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

Teaching Assistants



Yi-Hsuan Wu
吳奕萱



Zheng-Xin Weng
翁正欣



Wei-Da Chen
陳韋達



Hsin-Chi Chu
朱欣祈

How to Find Us?

- Office Hour (TAs)
 - Wed. 3:30-5:20pm at Delta 723/724
- Email (**Personal question only**)
 - Yi-Hsuan Wu: yhwu@datalab.cs.nthu.edu.tw
 - Zheng-Xin Weng: zxweng@datalab.cs.nthu.edu.tw
 - Wei-Da Chen: wdchen@datalab.cs.nthu.edu.tw
 - Hsin-Chi Chu: hccchu@datalab.cs.nthu.edu.tw
- Online Forum
 - iLms



If I have Question?

- Always Google first !
 - Learn how to google is important.
- If you try your best but still can't catch it.
 - Feel free to ask us on iLMS or office hour.



Team Up

- 2~4 people each team
 - 2 people is accepted if you can do as well as others.
- Please register your team here before **3/15 23:59**
 - Register form : <https://goo.gl/8DLTjD>
 - After that day we will match the rest student.

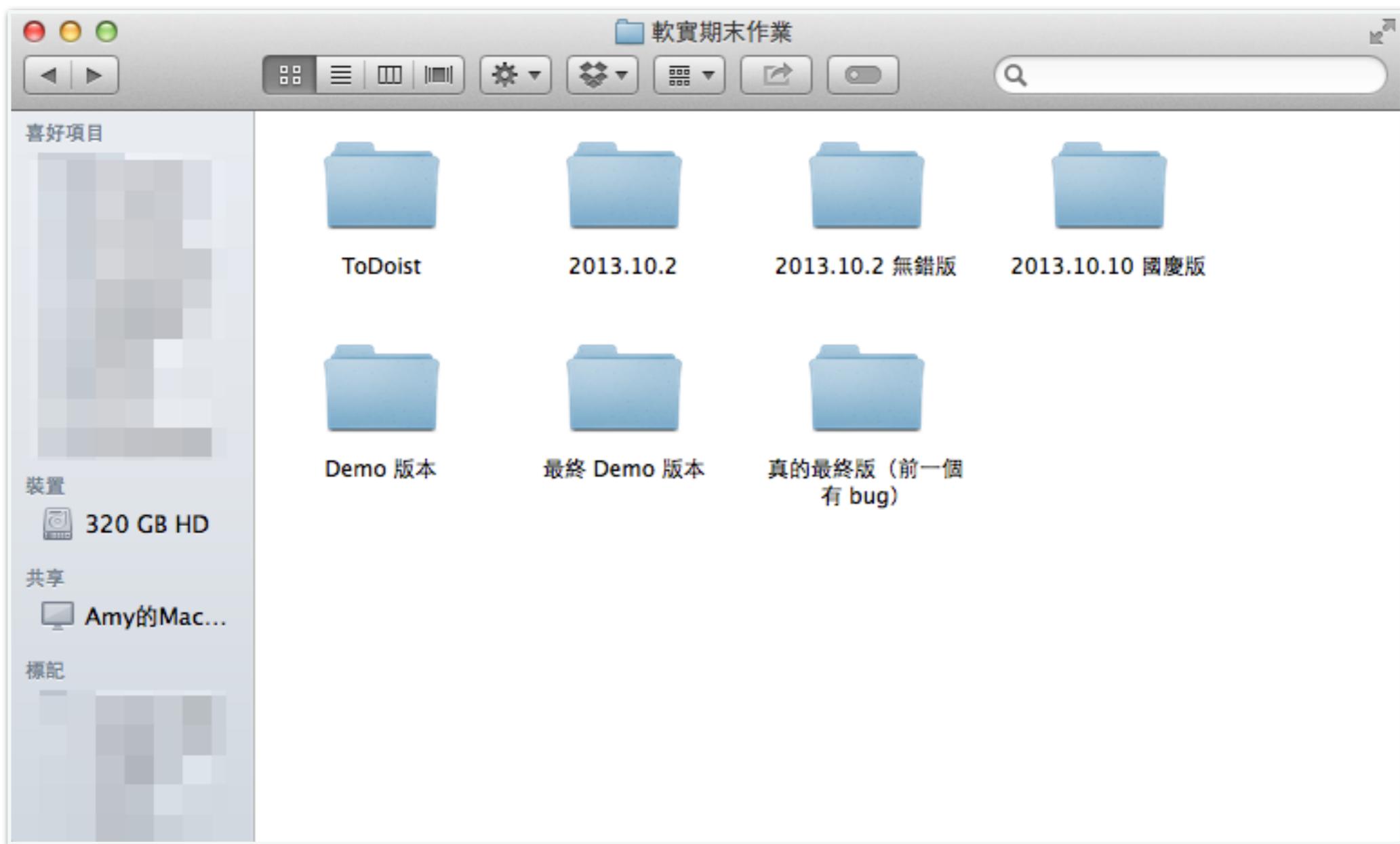
Outline

- General Rule
- Introduction to Git
 - Version control
 - Git Basics
 - Try Git!
 - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

Why use version control?

We want to track what we did and when we did it.

Students' VCS



How to work with others?



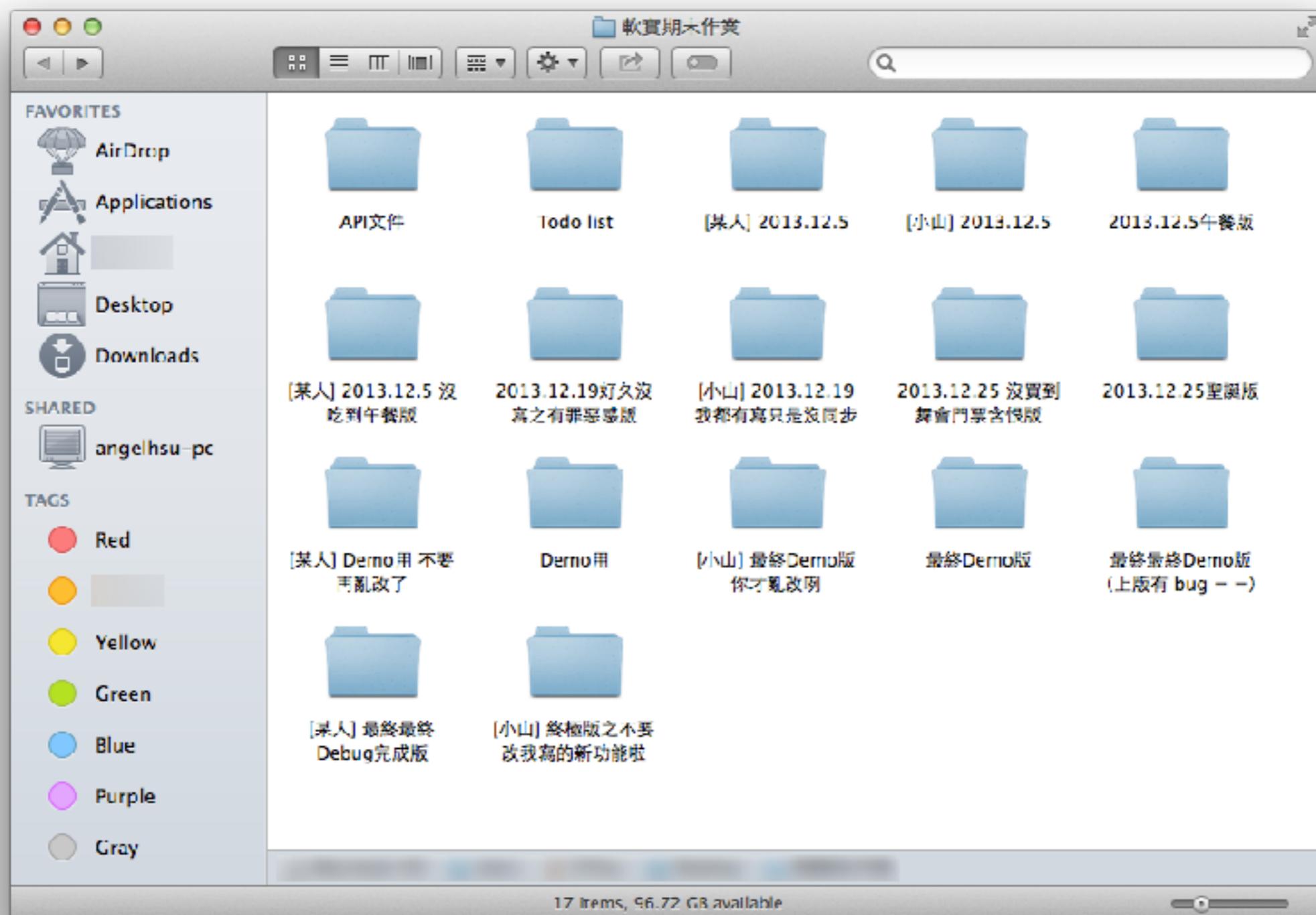
?



Dropbox

?

Dropbox VCS in Reality



Why use VCS?

- Managing your projects - tracking your files and modifications.
- Synchronization between modifications made by different developers.
- Revision history is still very helpful even if you work alone.

Outline

- General Rule
- Introduction to Git
 - Version control
 - **Git Basics**
 - Try Git!
 - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

Git



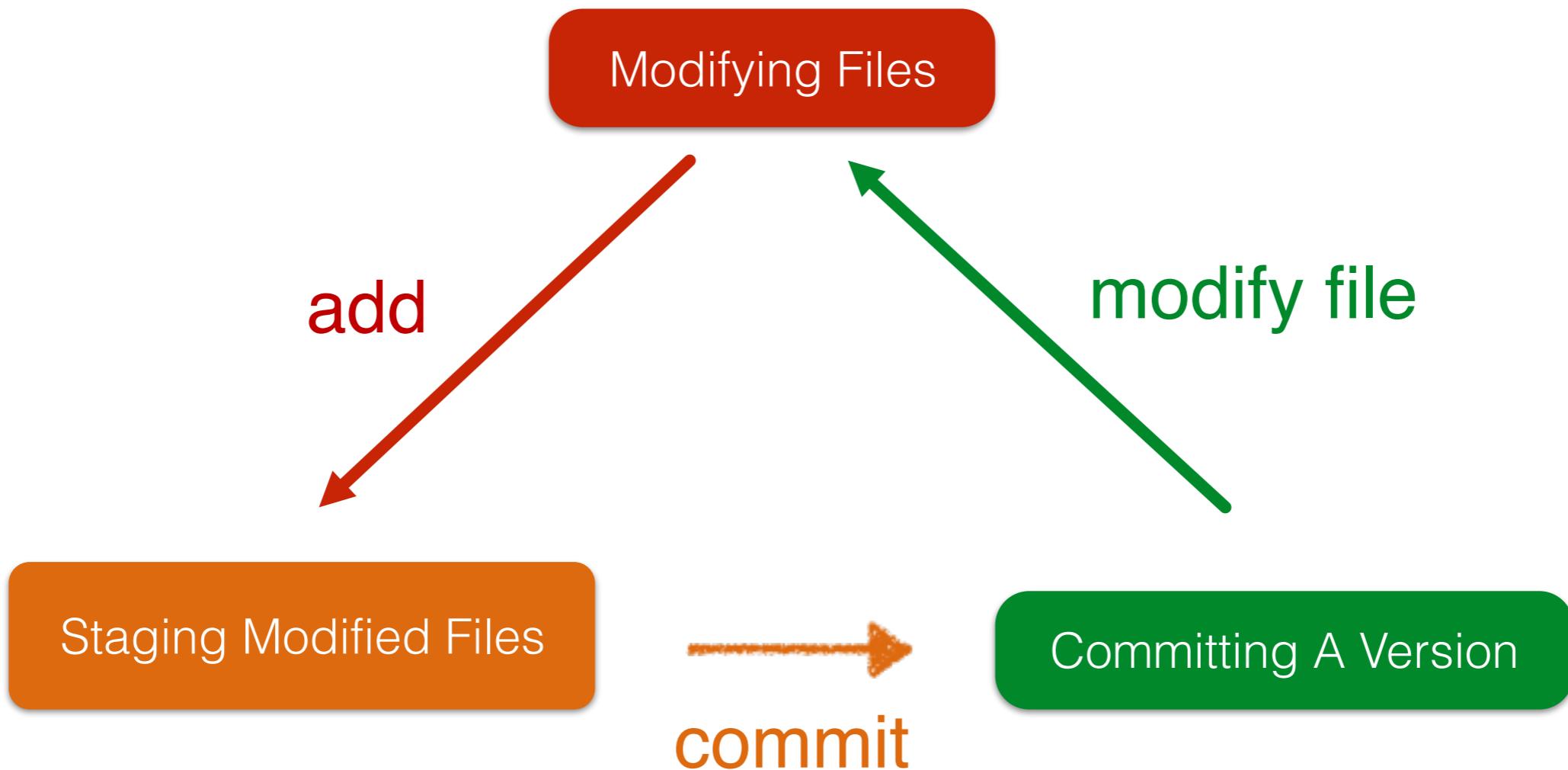
Git

- Git is a popular version control system which is
 - Fast
 - Easy to use
 - Distributed
- A git repository is a mini database that tracks your files.

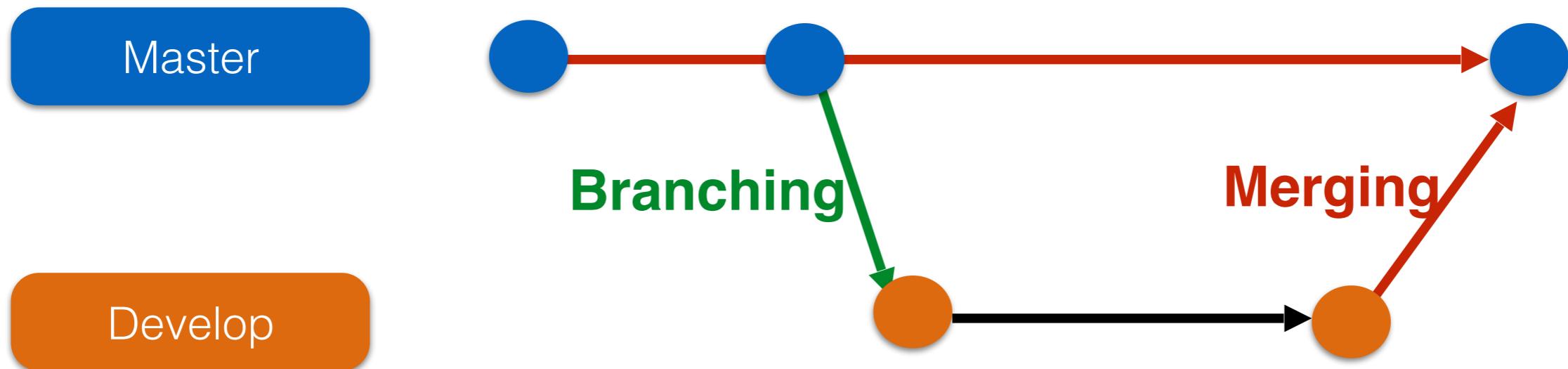
Git Workflow (1/2)

- With a local repository in your computer, you'll need following operations to make git track your work:
 - Create/modify files
 - Let git monitor the files by *adding* them to staging files.
 - Commit* your changes to and git will create snapshots (versions) of the files for you.

Git Workflow (2/2)



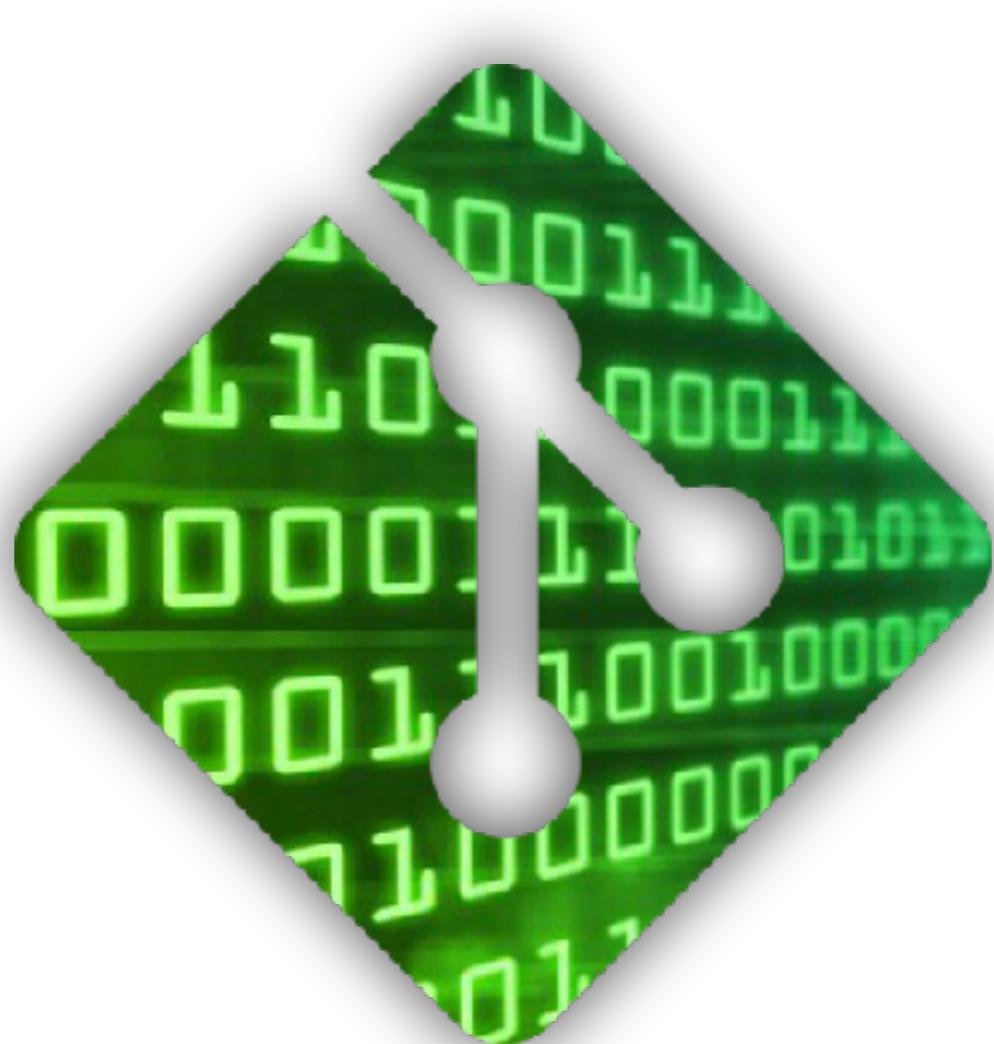
Git Branch



Outline

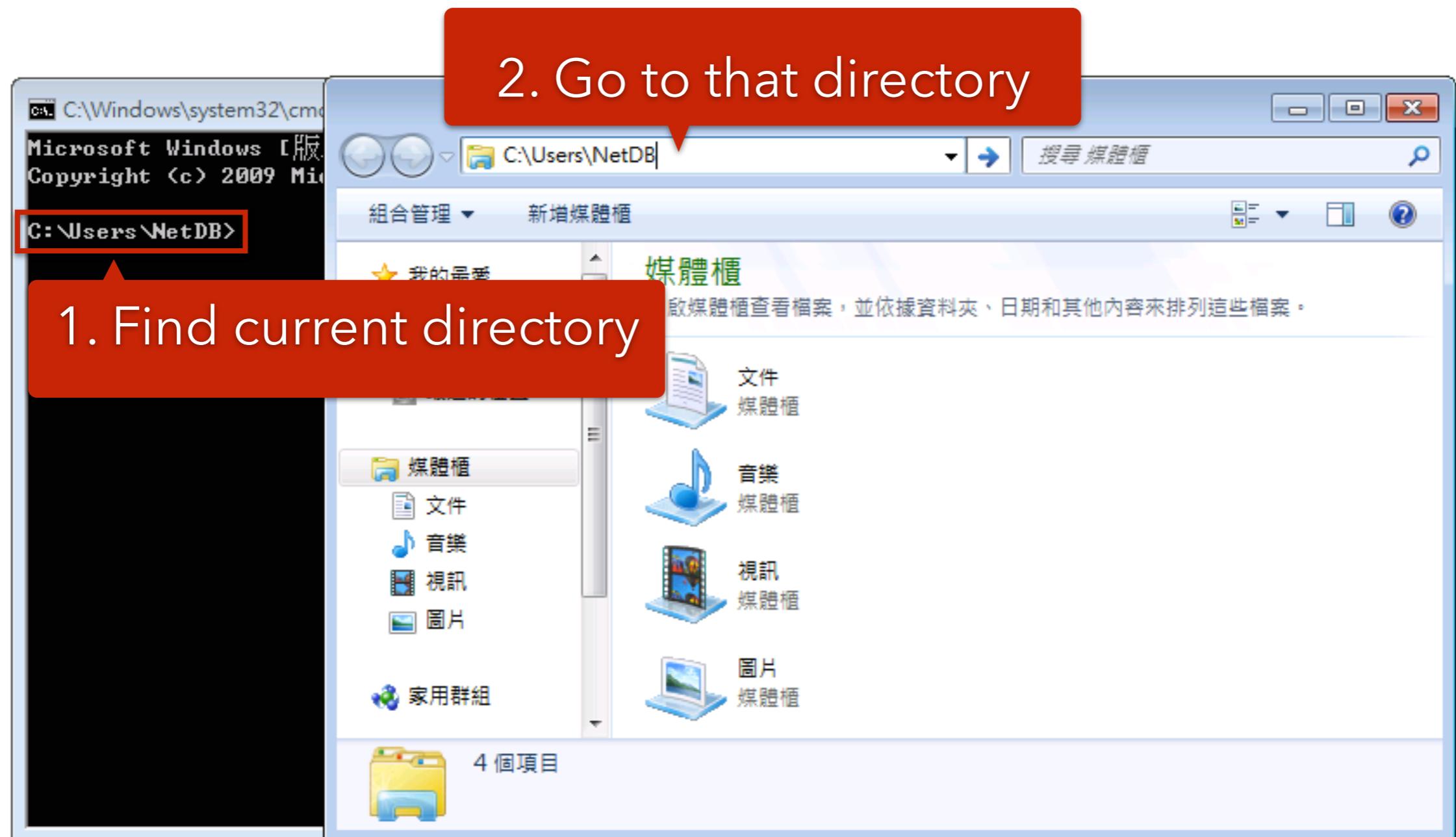
- General Rule
- **Introduction to Git**
 - Version control
 - Git Basics
 - Try Git!
 - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

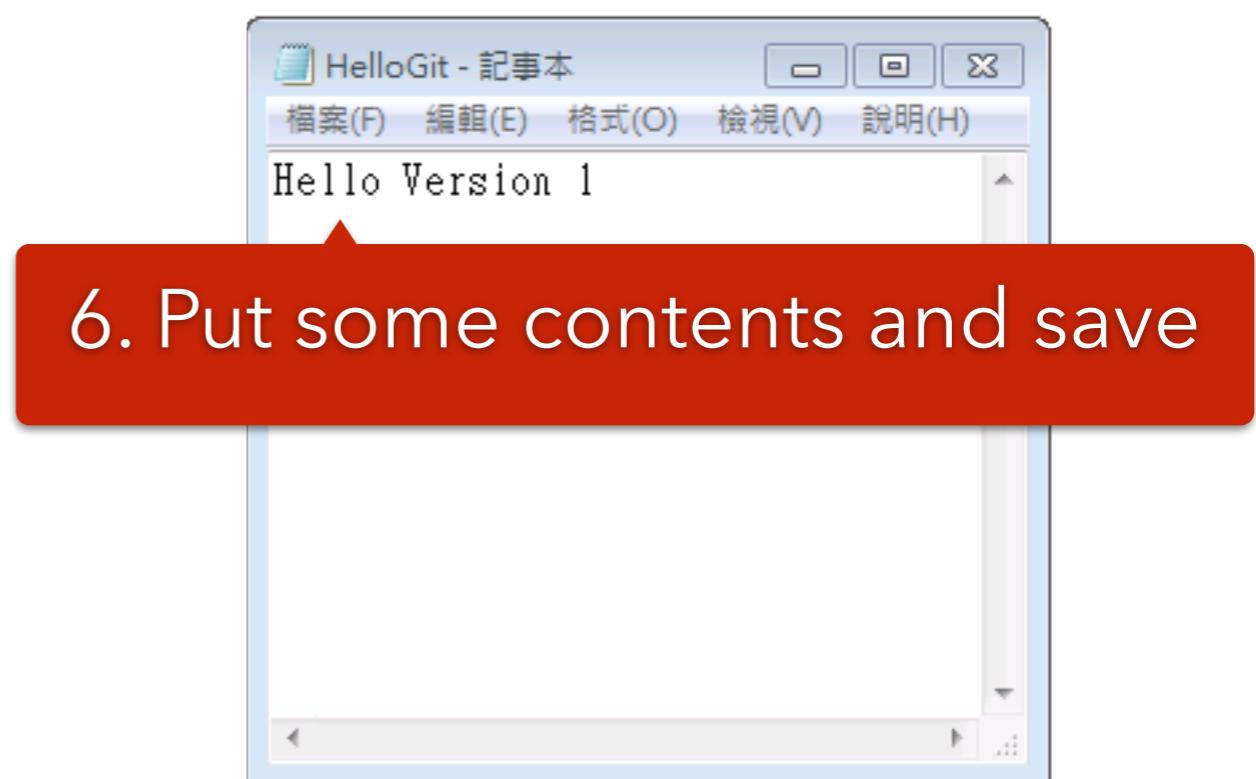
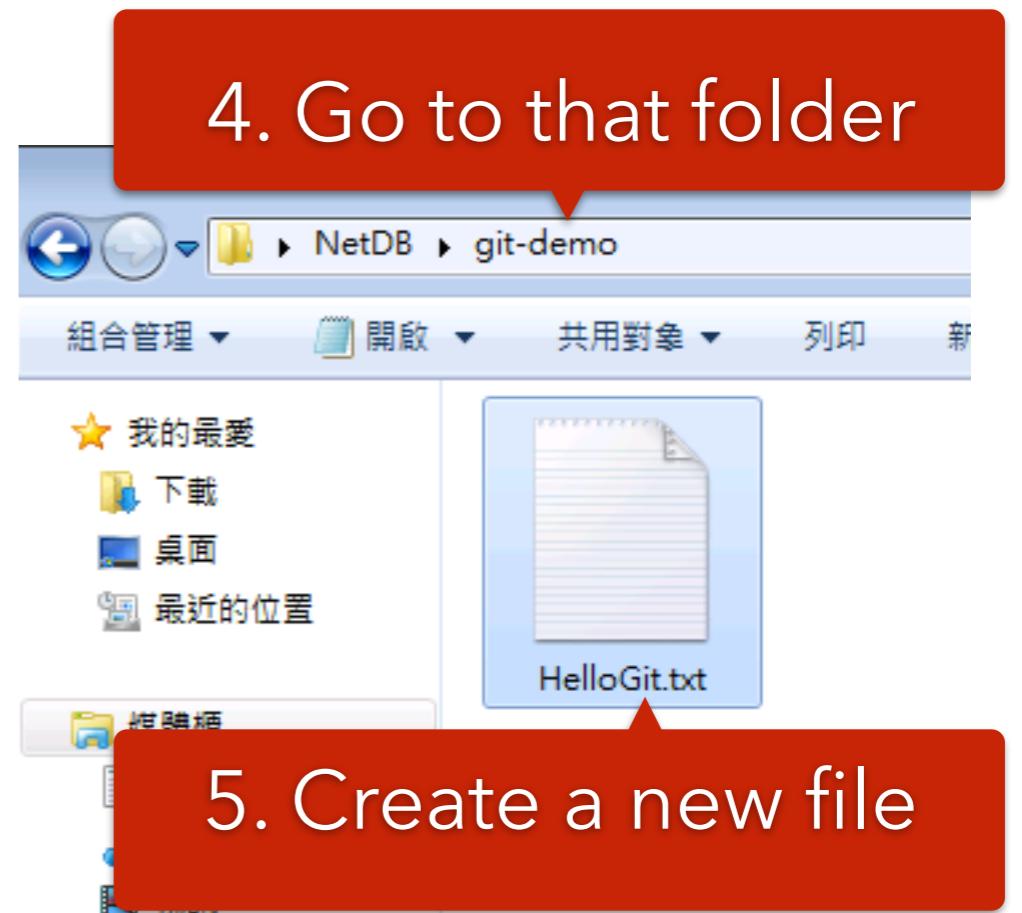
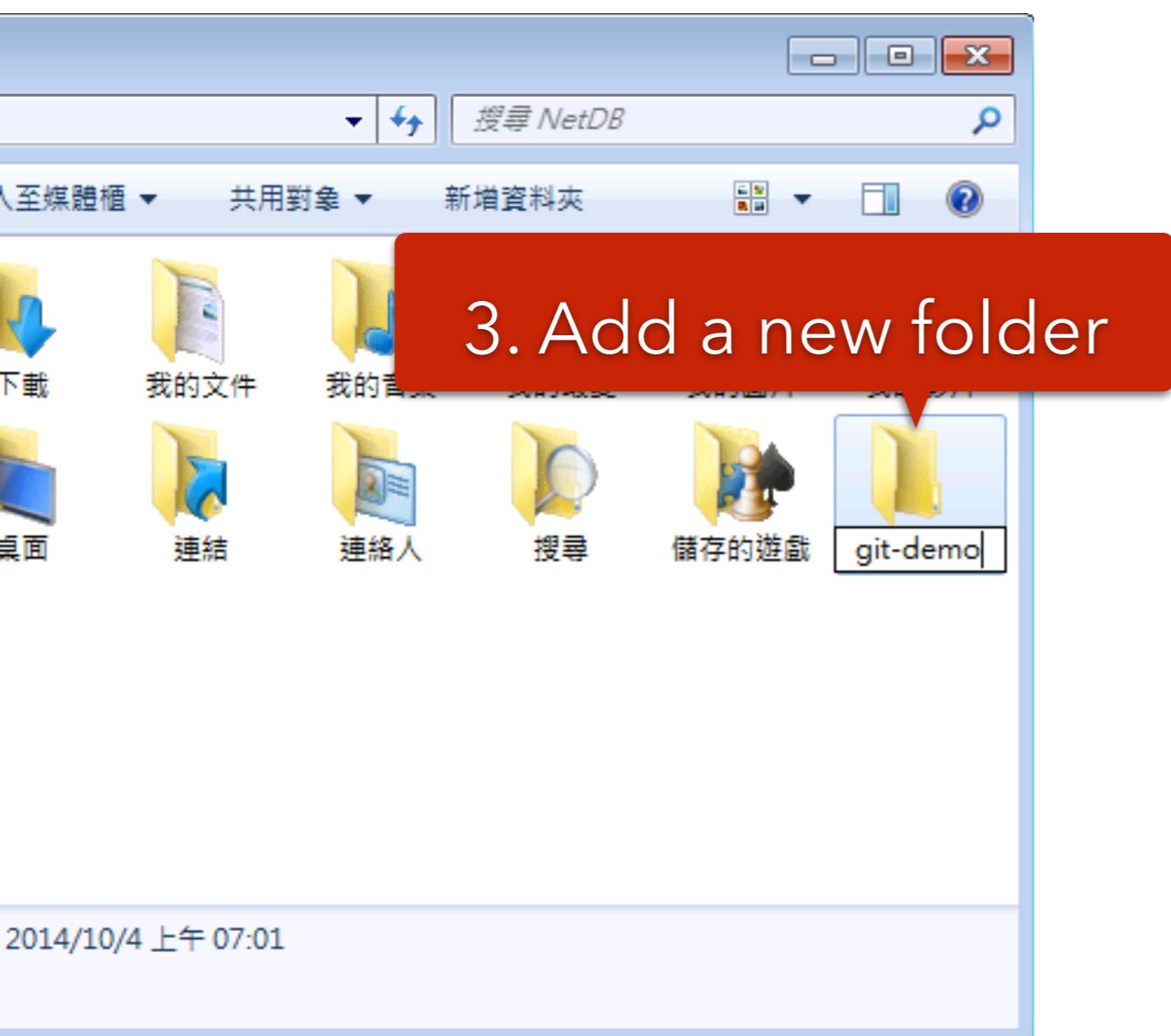
Be Professional

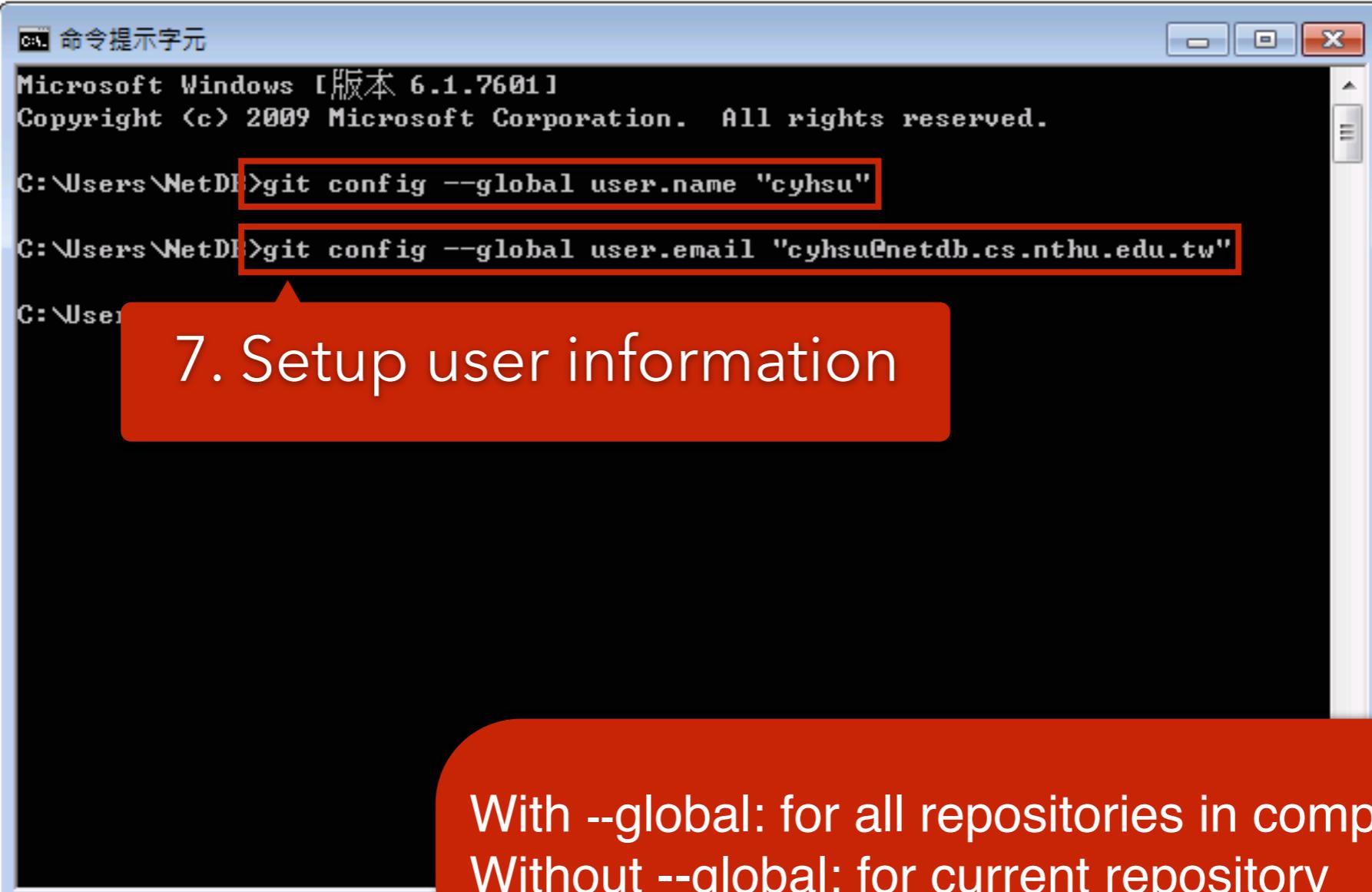


Basic Git Commands (1/2)

- **git init**
 - Initialize a repository at current directory.
- **git add [file_name]**
 - Add files to git repository and let git track them.
- **git commit -m "commit messages"**
 - Save the changes to the git repository and create snapshots of the files.
- **git checkout [version]**
 - Go to a specific version.







```
C:\ 命令提示字元  
Microsoft Windows [版本 6.1.7601]  
Copyright © 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\NetDB>git config --global user.name "cyhsu"  
C:\Users\NetDB>git config --global user.email "cyhsu@netdb.cs.nthu.edu.tw"  
C:\Users\NetDB>
```

7. Setup user information

With --global: for all repositories in computer
Without --global: for current repository

```
$ git config --global user.name "name"  
$ git config --global user.email "email"
```

命令提示字元
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\NetDB>cd git-demo
C:\Users\NetDB>git config user.name 'cyhsu'
C:\Users\NetDB>git config user.email "cyhsu@netdb.cs.nthu.edu.tw"
C:\Users\NetDB>cd git-demo
C:\Users\NetDB\git-demo>dir
 磁碟區 C 中的磁碟是 WIN7
 磁碟區序號: 187B-C5C9
C:\Users\NetDB\git-demo 的目錄
2014/10/04 上午 02:12 <DIR>
 .
 ..
 15 HelloGit.txt
 15 位元組
 6,944 位元組可用
C:\Users\NetDB\git-demo>git init
Initialized empty Git repository in C:/Users/NetDB/git-demo/.git/
C:\Users\NetDB\git-demo>

8. Go to "git-demo"

9. Show the files in "git-demo"

10. Initialize a Git repository

```
$ cd git-demo      # go to git-demo directory
$ dir              # list the files
$ git init         # initialize a repository
```

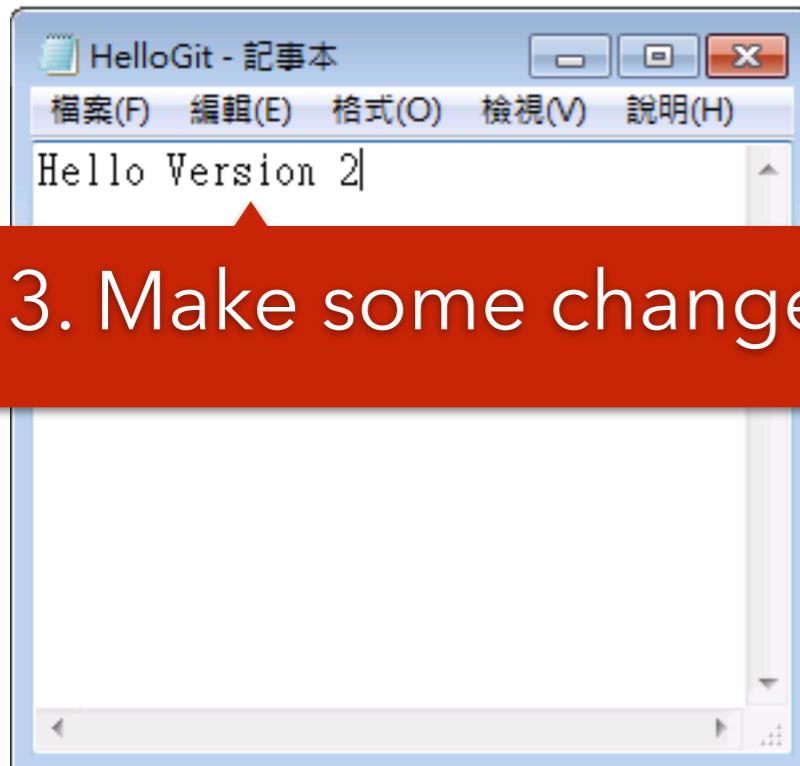
```
命令提示字元  
C:\Users\NetDB>cd git-demo  
  
C:\Users\NetDB\git-demo>dir  
磁碟區 C 中的磁碟是 WIN7  
磁碟區序號： 187B-C5C9  
  
C:\Users\NetDB\git-demo 的目錄  
  
2014/10/04 上午 07:17 <DIR> .  
2014/10/04 上午 07:17 <DIR> ..  
2014/10/04 上午 07:16 1 個檔案 15 HelloGit.txt  
15 位元組  
  
C:\Users\NetDB>git add HelloGit.txt  
  
C:\Users\NetDB\git-demo>git commit -m "version 1"  
[master (root-commit) b302d9c] version 1  
1 file changed, 1 insertion(+)  
create mode 100644 HelloGit.txt  
  
C:\Users\NetDB\git-demo>
```

11. Add HelloGit.txt to staging files

12. Commit your changes

```
# Add HelloGit.txt to staging files  
$ git add HelloGit.txt  
  
# Commit the changes to the repository  
# where "version 1" is the commit message  
$ git commit -m "version 1"
```

14. Add it and commit again



13. Make some changes and save

```
C:\Users\NetDB\git-demo>git add HelloGit.txt  
C:\Users\NetDB\git-demo>git commit -m "version 2"  
[master e134c84] version 2  
1 file changed, 1 insertion(+), 1 deletion(-)  
git-demo>
```

A screenshot of a terminal window. The command 'git add HelloGit.txt' is highlighted with a red box. The command 'git commit -m "version 2"' is also highlighted with a red box. The output shows a commit message '[master e134c84] version 2' with '1 file changed, 1 insertion(+), 1 deletion(-)'. The prompt 'git-demo>' is at the bottom.

15. View your versions

Version ID

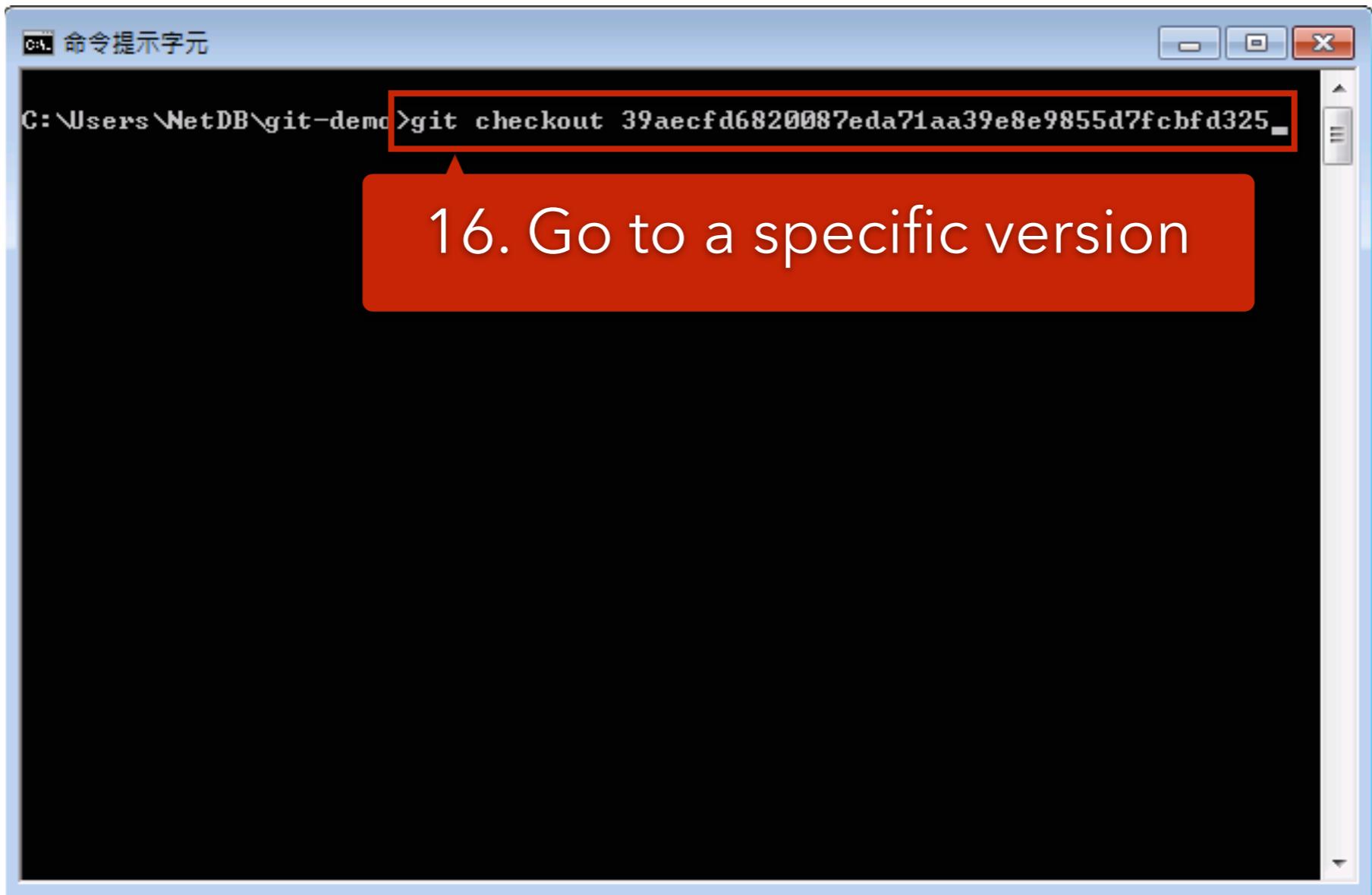
```
C:\Users\NetDB\git-demo>git log
commit e134c845df593f1451c4e9e6c874ddef6df42a76
Author: cyhsu <cyhsu@netdb.cs.nthu.edu.tw>
Date:   Sat Oct 4 08:09:55 2014 +0800

    version 2

commit 39aecfd6820087eda71aa39e8e9855d7fcfd325
Author: cyhsu <cyhsu@netdb.cs.nthu.edu.tw>
Date:   Sat Oct 4 08:09:16 2014 +0800

    version 1
```

```
# Show the versions you've created so far
$ git log
```



```
# Go to a specific version  
$ git checkout {version_id}
```

**LIFE IS
TOO SHORT
TO TYPE THAT
VERSION ID!**

which is 40 characters long...

version ID

A screenshot of a Windows Command Prompt window titled "命令提示字元". The command entered is "git log --oneline". The output shows two commits:

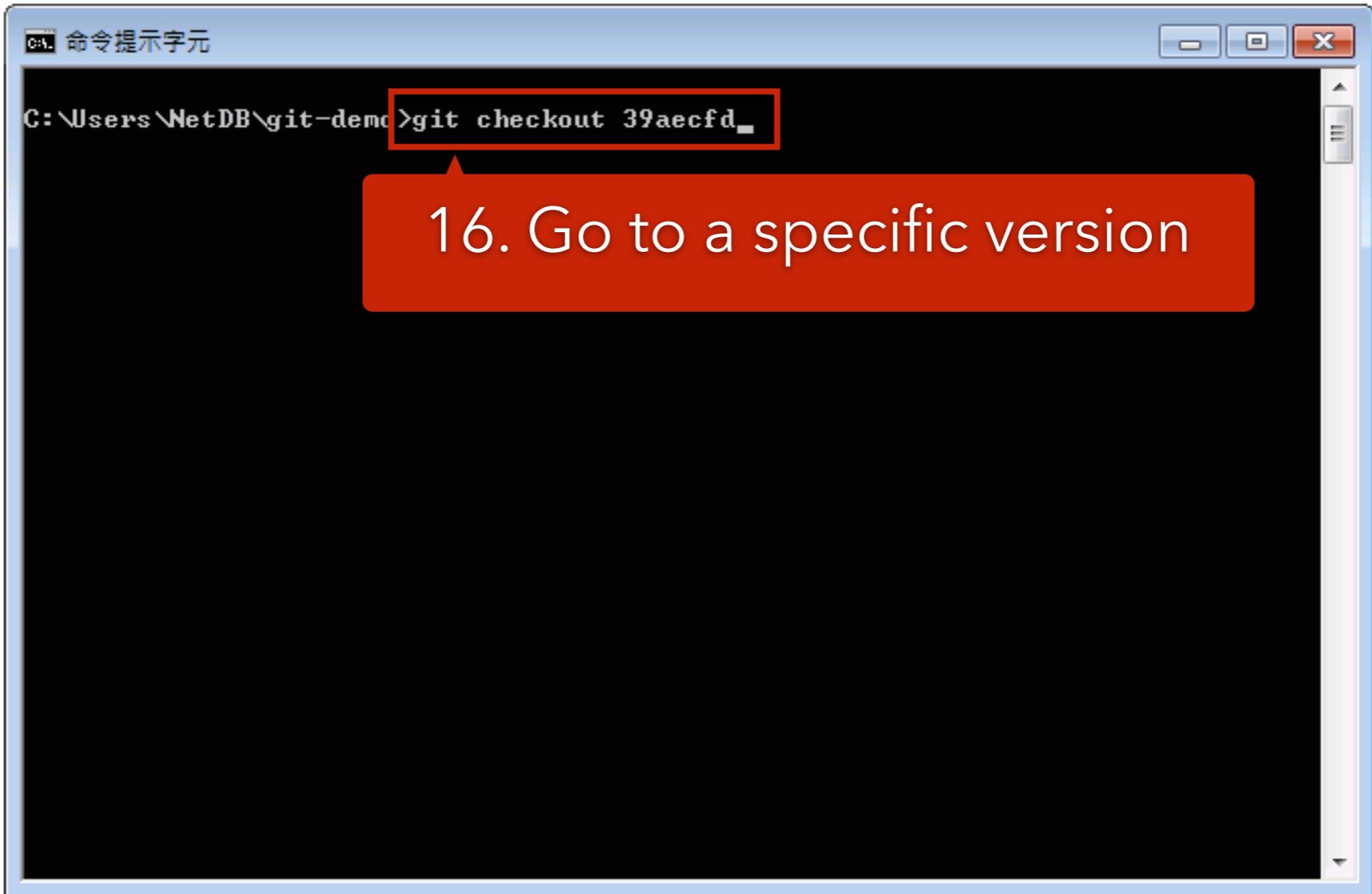
```
C:\Users\NetDB\git-demo>git log --oneline
e134c84 version 2
39aecfd version 1
```

The first commit hash "e134c84" is highlighted with a red rectangle. A blue arrow points from the text "version ID" to this highlighted area. A red callout box contains the text "15. Show versions with short version ID". Another blue arrow points from the bottom of this callout box to the text "56% shorter!".

15. Show versions with short version ID

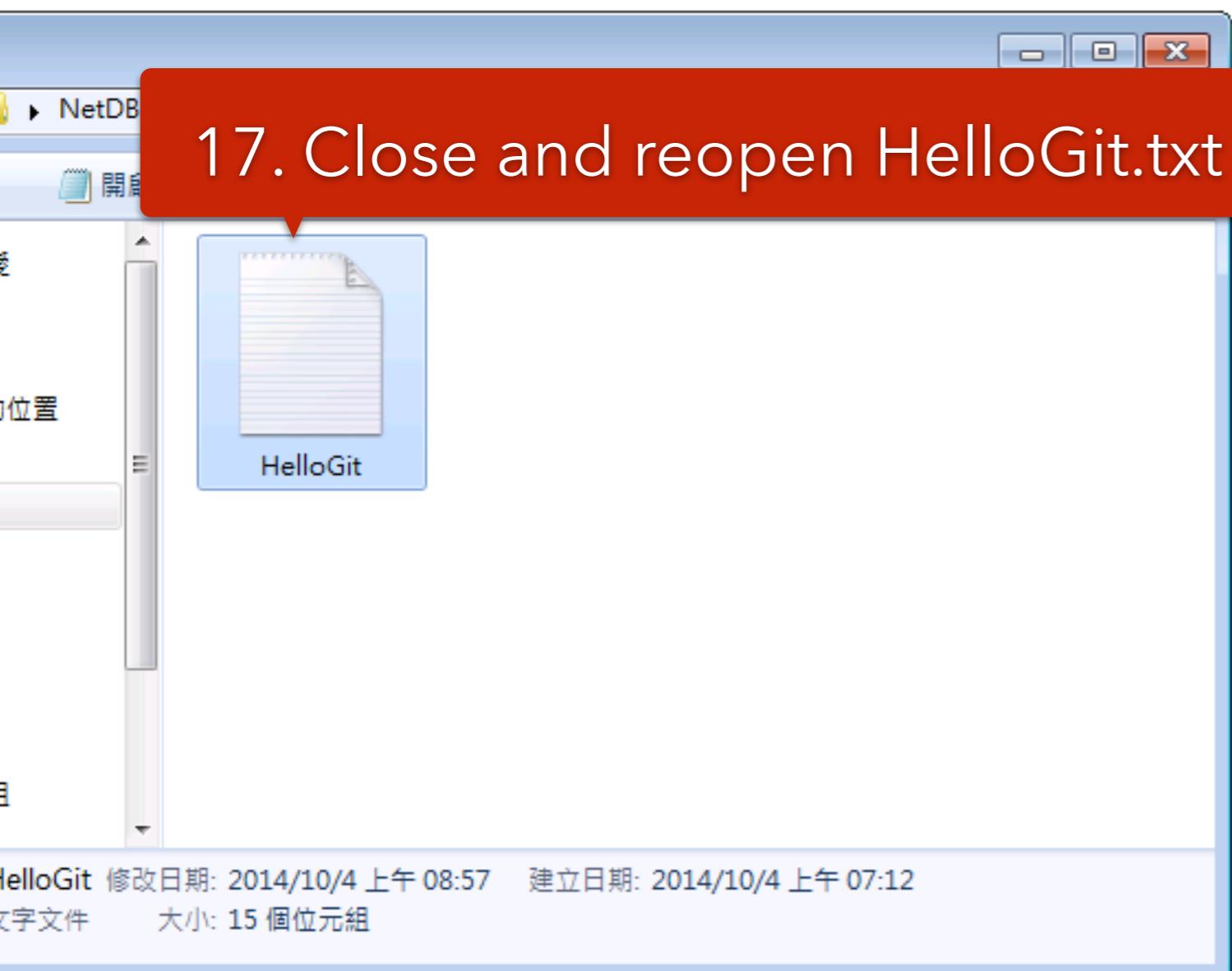
56% shorter!

```
# Show versions with short version id
$ git log --oneline
```



```
# Go to a specific version.  
# In fact, you only need to type  
# the first 5 characters.  
$ git checkout {short_version_id}
```

17. Close and reopen HelloGit.txt



18. Back to the version 1!

Try yourself (1/2)

- Branching steps
 - Creating a new branch

`git branch [branch name]`

- Checking out the branch

`git checkout [branch name]`

Try yourself (2/2)

- Merging steps
 - Checking out a branch to merge

`git checkout [branch 1 name]`

- Merging another branch

`git merge [branch 2 name]`

Outline

- General Rule
- Introduction to Git
 - Version control
 - Git Basics
 - Try Git!
 - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

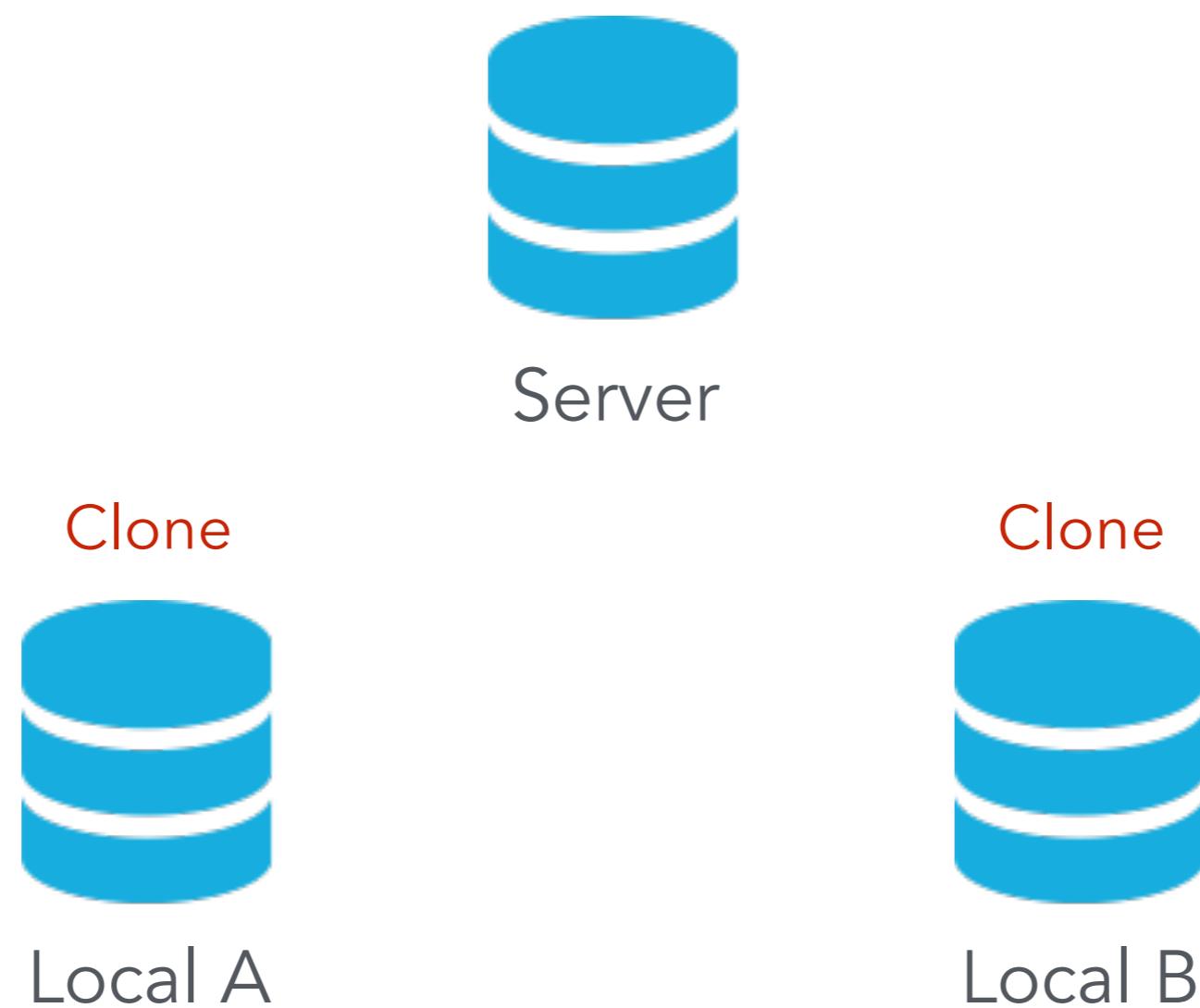
Collaboration

- To work with others using git, you'll need a server that store the repository.
- Git is distributed, which means
 - Everyone can store a copy of the repository downloaded from the server to their computer and do their jobs independently.

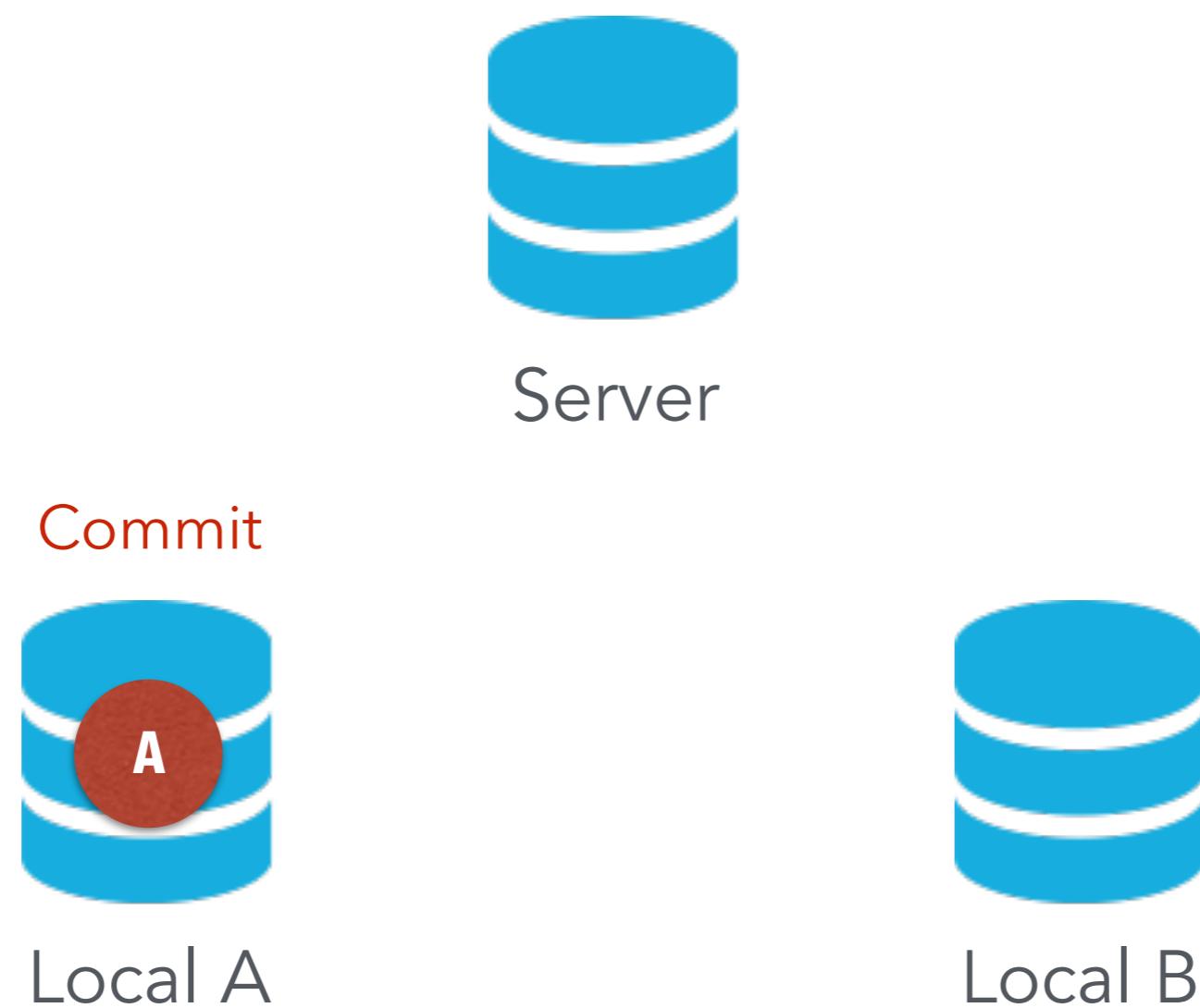
Collaboration Workflows (1/2)

1. If you don't have the project, *clone* (download) the repository from the server.
2. Do your work and commit the changes at local. Once done, *push* (upload) the repository to the server.
3. If someone else modified the project, you can *pull* (sync) the repository to get the updated project.
4. Repeat 2 and 3.

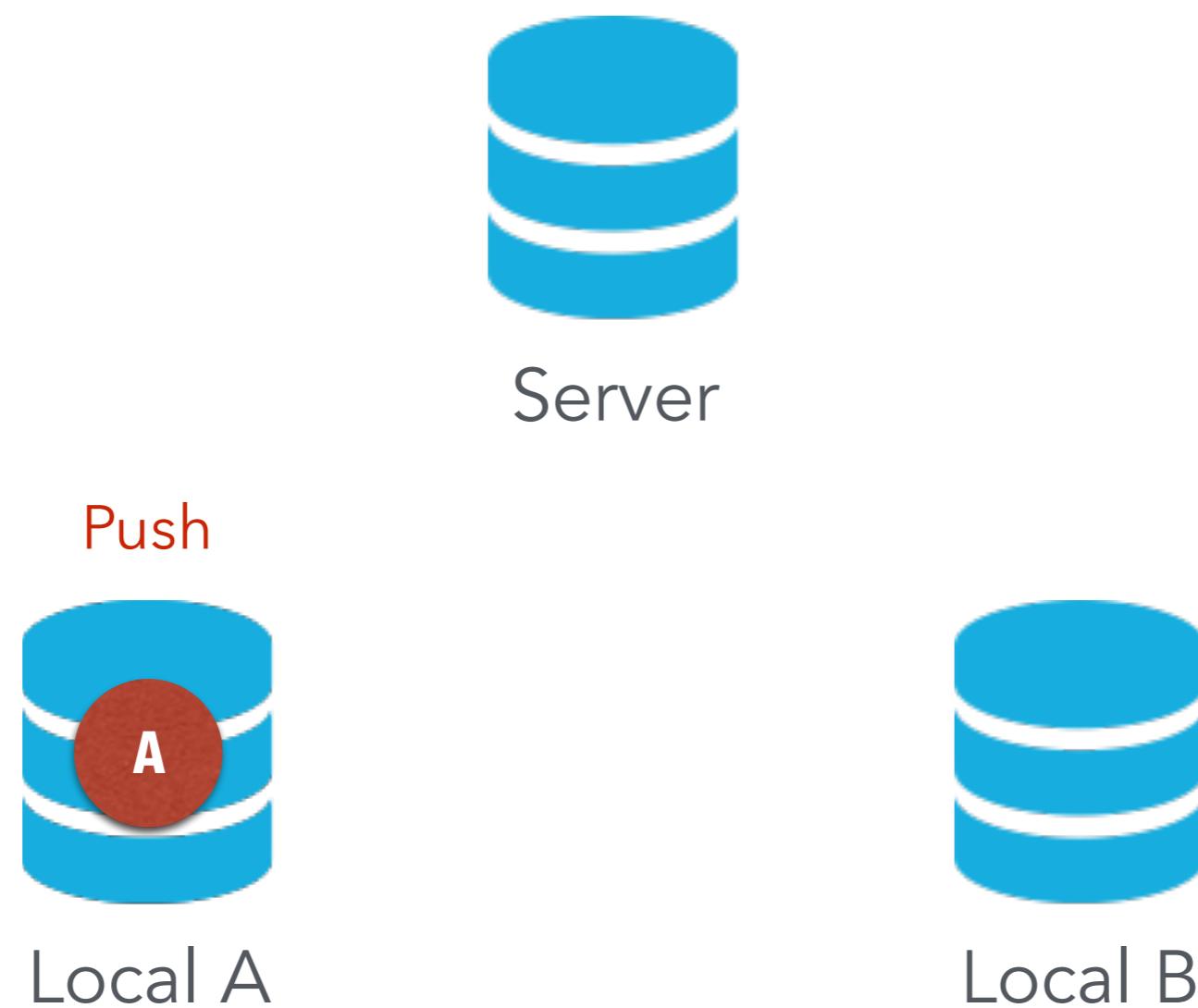
Collaboration Workflows (2/2)



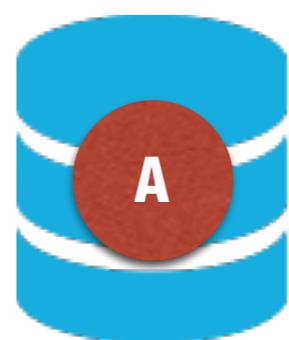
Collaboration Workflows (2/2)



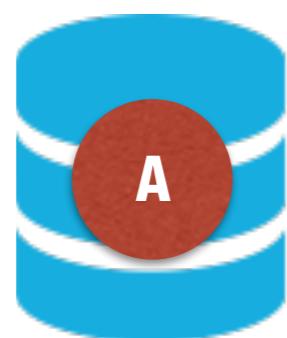
Collaboration Workflows (2/2)



Collaboration Workflows (2/2)



Server



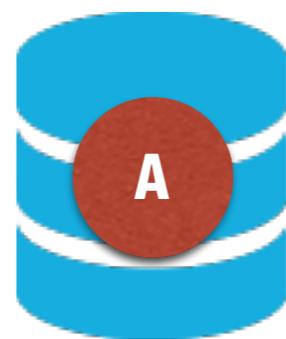
Local A



Local B

Pull

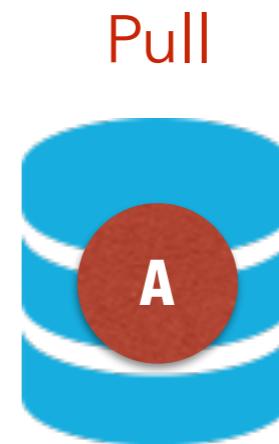
Collaboration Workflows (2/2)



Server



Local A



Local B

Pull

Why Authentication Failed?

Collaboration Workflow

- If you tried to clone the code template from a server and want to push the modified file.
 - You will get authentication failed.
 - It's because it was a project of others, which means you are not able to save the changes back to the server.
- So, how can I copy a project from others on a open source platform like Github?

Introducing
Fork



Original Project



Forked Project



Forked from Red

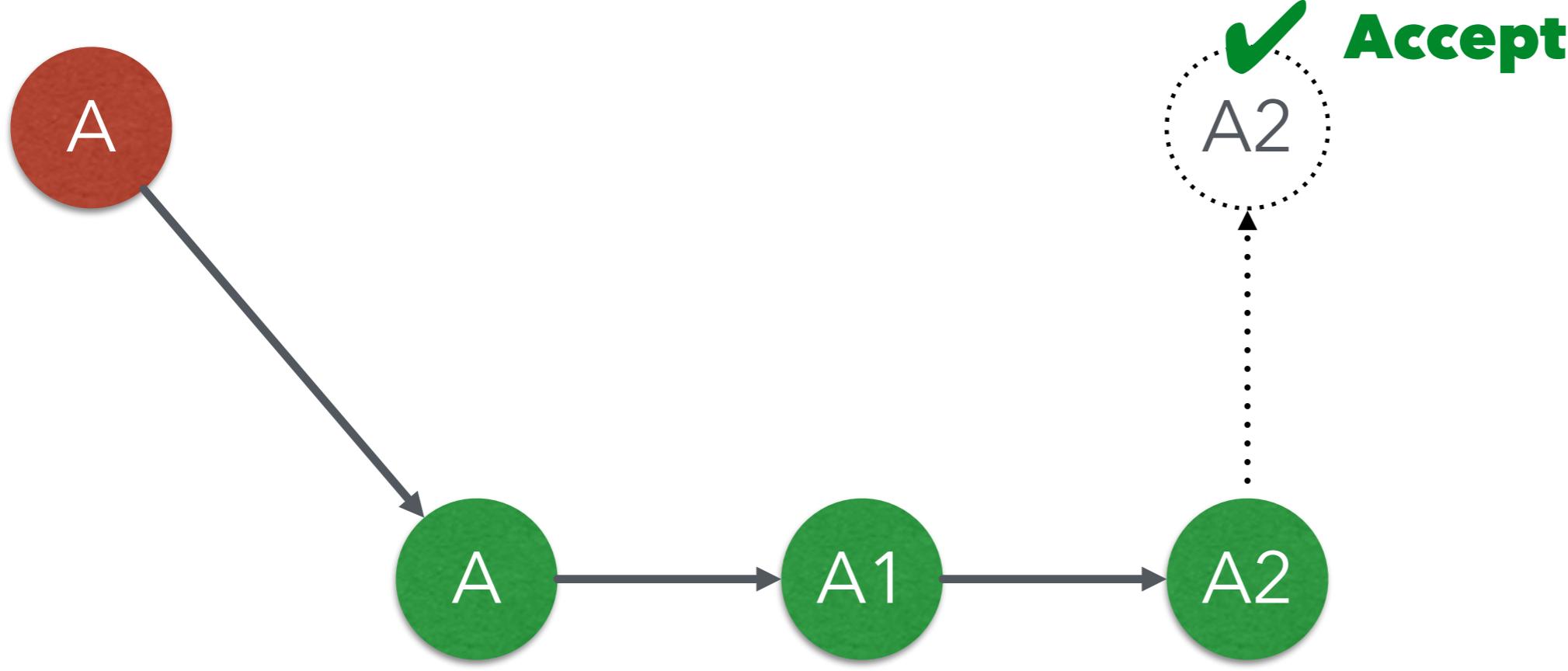


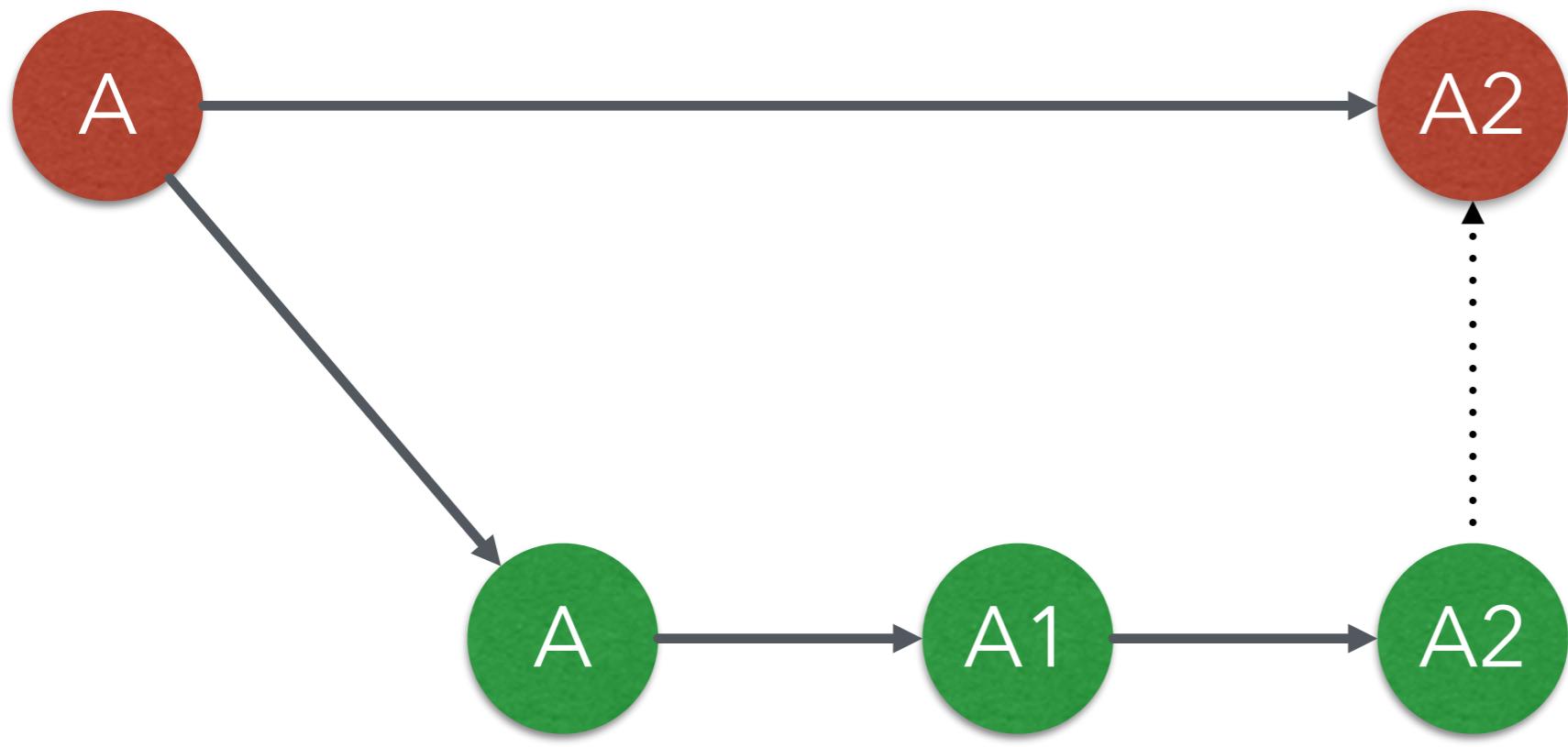
Commit

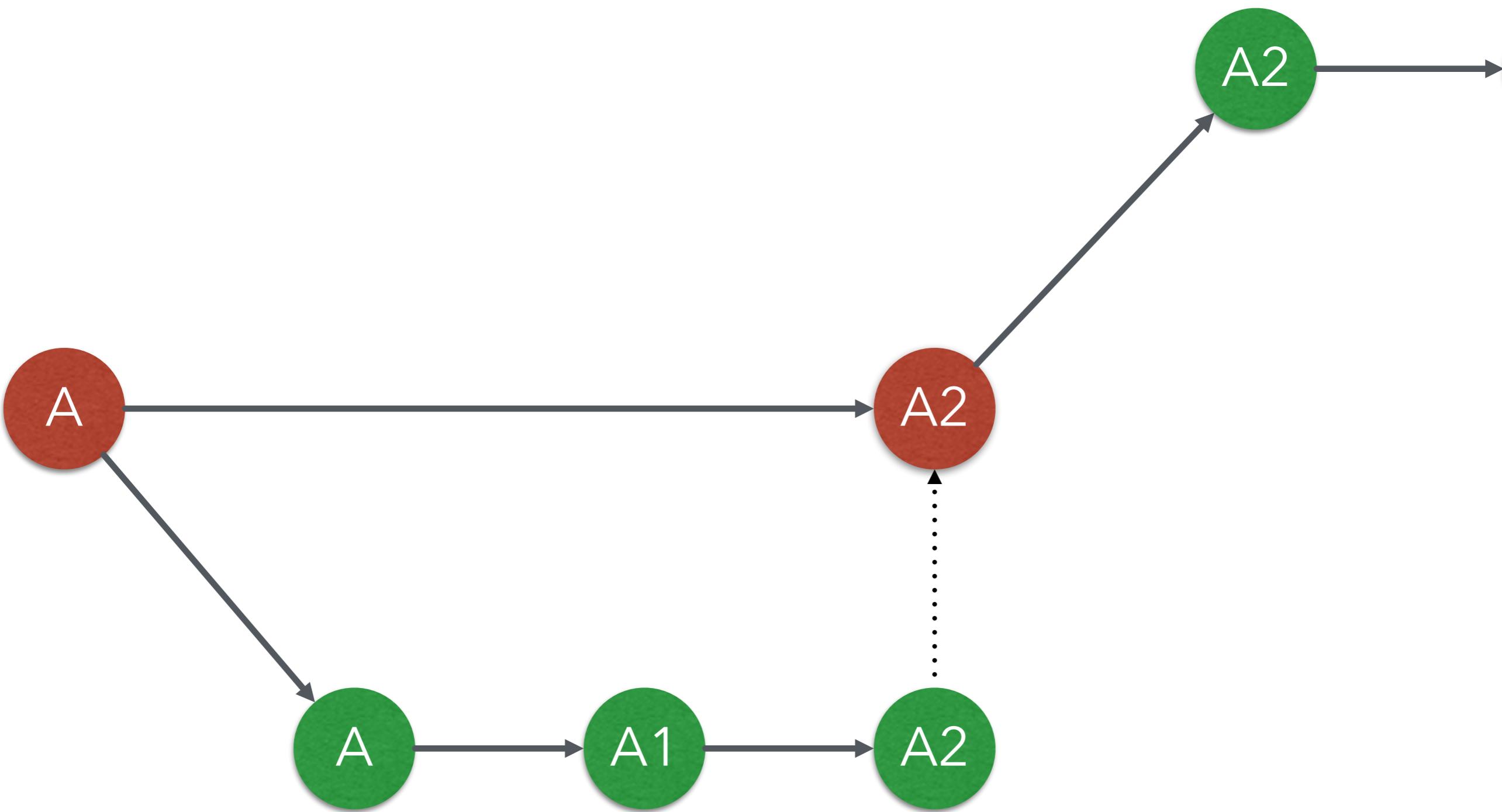


Commit

Open a Merge Request







Git Collaboration Workflow

1. *Fork* a repository to make a copy of it.
2. *Clone* the repository you forked to your workspace.
3. Do your work and *commit* the changes in your workspace.
4. *Push* the repository to the server to synchronize them.
5. Open a *merge request* to origin repository .

Basic Git Commands (2/2)

- **git clone [url]**
 - Clone a repository from remote server
- **git push [url] [branch-name]**
 - Push committed file to remote server

Outline

- General Rule
- Introduction to Git
 - Version control
 - Git Basics
 - Try Git!
 - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

Gitlab

- We have created account for you
- Account: student ID (e.g. 106012345)
- Password: webapp+studentID (e.g. webapp106012345)

Workflow

- For each lab, you should follow the workflow below
 1. Fork our template repository on Gitlab
 2. Clone the **forked** repository to your computer
 3. Finish your lab
 4. Commit in your computer
 5. Push to Gitlab
 6. Send merge request of **your branch** to our template repository

Workflow

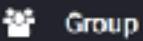
- For each lab, you should follow the workflow below
 1. Fork our template repository on Gitlab
 2. Clone the **forked** repository to your computer
 3. Finish your lab
 4. Commit in your computer
 5. Push to Gitlab
 6. Send merge request of **your branch** to our template repository

You can access course projects in [this group](#)



GitLab

[Go to dashboard](#)



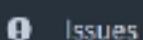
Group



Activity



Milestones



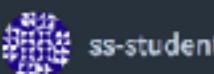
Issues 0



Merge Requests 0



Members



ss-student

courses-software-studio-2017-spring

Search 0 + ⌂

courses-software-studio-2017-spring ⓘ

@courses-software-studio-2017-spring

All Projects Filter by name Last updated ▾

S submission-exercise

...



GitLab

Go to group

Project

Activity

Files

Commits

Graphs

Milestones

Issues 0

Merge Requests 0

Labels

Wiki

Forks

courses-software-studio-2017-spring / submission-exercise ▾

Search

0 + ↗



S

submission-exercise 🛡

[Star](#)

0

[Fork](#)

0

HTTP ▾

<https://shwu10.cs.nthu.edu.tw/>

Global ▾

1. Click to fork

1 commit

109 branches

0 tags

0.48 MB

b1e0571c add README · about 4 hours ago by realwei

Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting a lab or a project.

Try It !!

1. Fork this project.
2. Clone the project from GitLab to your local environment.
3. Add a new file and write something.
4. Commit your work.
5. Push your work to GitLab.
6. Open a merge request to the original repository.
 - Source branch: Your working branch.
 - Target branch: The branch with your student id. (e.g. [104062987](#))
 - Title: **Student ID Submission** (e.g. [104062987 Submission](#)).
 - You can choose arbitrary branch if your branch doesn't exist.



ss-student

<



GitLab

Go to dashboard

Project

Activity

Files

Commits

Graphs

Milestones

Issues 0

Merge Requests 0

Members

Labels

Wiki

Forks

Settings

Software Studio Student / submission-exercise

Search

0 +

2. Check if this repository is under your account

S

3. Go to Settings

submission-exercise 🔒

Forked from courses-software-studio-2017-spring

Star

0

HTTP

https://shwu10.cs.nthu.edu.tw/ss-student/



Global

1 commit

109 branches

0 tags

0.09 MB

Add Changelog

Add License

Add Contribution guide

b1e0571c add README · about 5 hours ago by realwei

Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting a lab or a project.

Try It !!

1. Fork this project.
2. Clone the project from GitLab to your local environment.
3. Add a new file and write something.
4. Commit your work.
5. Push your work to GitLab.
6. Open a merge request to the original repository.
 - Source branch: Your working branch.
 - Target branch: The branch with your student id. (e.g. 104062987)
 - Title: Student ID Submission (e.g. 104062987 Submission).



<



Go to project

Project Settings

Groups

Deploy Keys

Webhooks

Services

Protected Branches

Project settings

Project name

submission-exercise

Project description

(optional)

Default Branch

4. Set project to private

Visibility Level (?)

Private

Project access must be granted explicitly to each user.

Internal

The project can be cloned by any logged in user.

Tags

Separate tags with commas.

Features:

Issues

Lightweight issue tracking system for this project

Merge Requests

Submit changes to be merged upstream

Builds

Test and deploy your changes before merge

Wiki

Pages for project documentation

Snippets





GitLab

[Go to project](#)[Project Settings](#)[Groups](#)[Deploy Keys](#)[Webhooks](#)[Services](#)[Protected Branches](#)

Software Studio Student / submission-exercise · Settings

 Search[0](#) [+](#) [≡](#)

Internal

The project can be cloned by any logged in user.

Tags

Separate tags with commas.

Features:

Issues

Lightweight issue tracking system for this project

Merge Requests

Submit changes to be merged upstream

Builds

Test and deploy your changes before merge

Wiki

Pages for project documentation

Snippets

Share code pastes with others out of git repository

Project avatar:

You can upload a project avatar here

[Choose File ...](#) [File name...](#)

5. Scroll down and save changes

[Save changes](#)



ss-student

Workflow

- For each lab, you should follow the workflow below
 1. Fork our template repository on Gitlab
 2. Clone the **forked** repository to your computer
 3. Finish your lab
 4. Commit in your computer
 5. Push to Gitlab
 6. Send merge request of **your branch** to our template repository



GitLab

Go to dashboard

Project

Activity

Files

Commits

Graphs

Milestones

Issues 0Merge Requests 0

Members

Labels

Wiki

Forks

Settings

Software Studio Student / submission-exercise ▾

Search

0 + ↗



S

submission-exercise 🔒

Forked from courses-software-studio-2017-spring

★ Star

0

2. Copy the link

HTTP ▾

<https://shwu10.cs.nthu.edu.tw/ss-student/>

Global ▾

1. Choose HTTP

branches

0 tags

0.09 MB

Add Changelog

Add License

Add Contribution guide

ble0571c add README · about 5 hours ago by realwei

Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting a lab or a project.

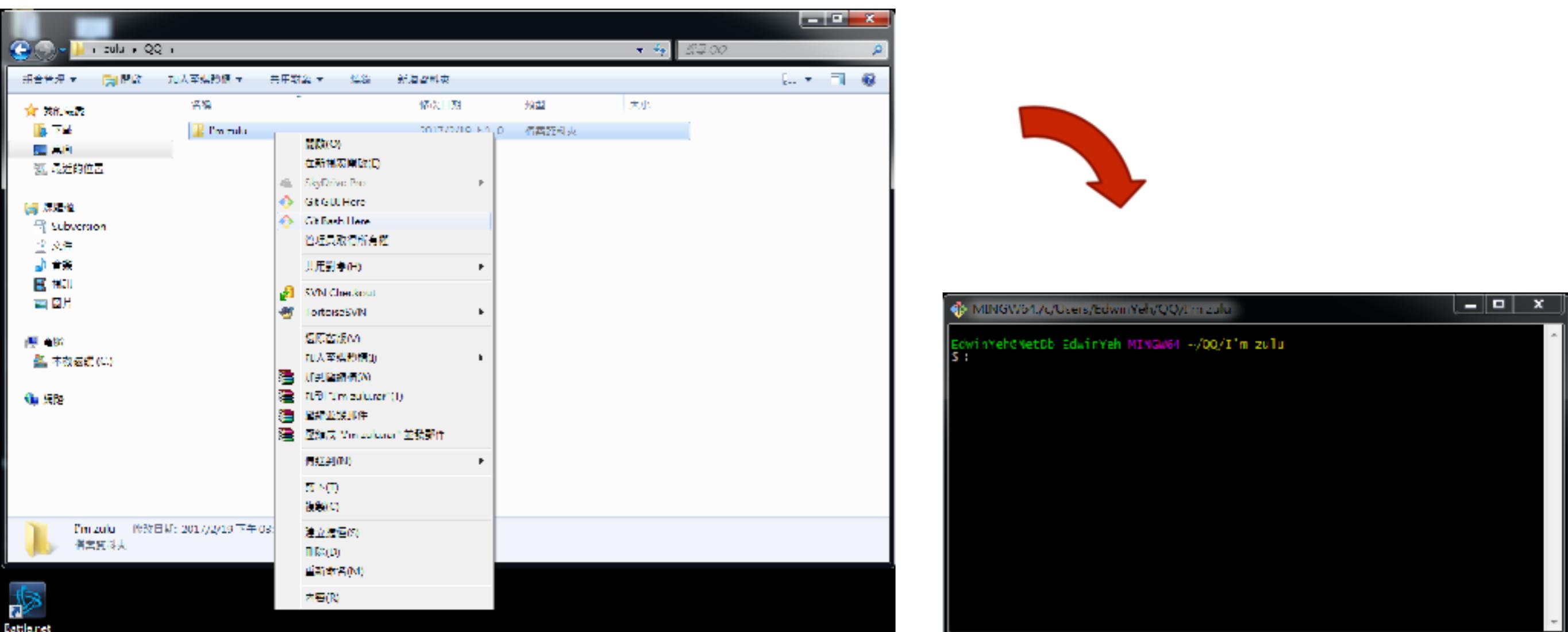
Try It !!

1. Fork this project.
2. Clone the project from GitLab to your local environment.
3. Add a new file and write something.
4. Commit your work.
5. Push your work to GitLab.
6. Open a merge request to the original repository.
 - Source branch: Your working branch.
 - Target branch: The branch with your student id. (e.g. [104062987](#))
 - Title: **Student ID Submission** (e.g. [104062987 Submission](#)).



<

If You use Windows



3. Create a folder to put your repos

```
~/SS-Projects ➔ git clone https://shwu10.cs.nthu.edu.tw/ss-student/submission-exercise.git
Cloning into 'submission-exercise'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0)
Unpacking objects: 100% (3/3), done.
~/SS-Projects ➔ ls
submission-exercise
~/SS-Projects ➔
```

4. Type "git clone {URL}"

5. The repo has been successfully cloned

Workflow

- For each lab, you should follow the workflow below
 1. Fork our template repository on Gitlab
 2. Clone the **forked** repository to your computer
 3. Finish your lab
 4. Commit in your computer
 5. Push to Gitlab
 6. Send merge request of **your branch** to our template repository

```
~/SS-Projects/submission-exercise [master] vim lab1.js
~/SS-Projects/submission-exercise [master] git add -A
~/SS-Projects/submission-exercise [master +] git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   lab1.js
```

1. -A means all files

```
~/SS-Projects/submission-exercise [master +] git commit -m "Finish lab1"
[master c1acaf4] Finish lab1
 1 file changed, 1 insertion(+)
 create mode 100644 lab1.js
~/SS-Projects/submission-exercise [master]
```

2. Check if your file is added to git

3. Commit your changes

```
~/SS-Projects/submission-exercise ➤ master ➤ vim lab1.html
~/SS-Projects/submission-exercise ➤ master ➤ git add -A
~/SS-Projects/submission-exercise ➤ master + ➤ git commit -m "Finish lab1"
[master 8a603d9] Finish lab1
Committer: Real Wei <realwei@Realweis-MBP.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

git config --global user.name "Your Name"
git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with

git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lab1.html
~/SS-Projects/submission-exercise ➤ master ➤
```

If you see these message, type
git config --global user.name "{name}"
git config --global user.email "{email}"

{email} is the email you use on gitlab

Workflow

- For each lab, you should follow the workflow below
 1. Fork our template repository on Gitlab
 2. Clone the **forked** repository to your computer
 3. Finish your lab
 4. Commit in your computer
 5. Push to Gitlab
 6. Send merge request of **your branch** to our template repository

```
~/SS-Projects/submission-exercise ➜ ⌂ master ➜ git push -u origin master  
Counting objects: 6, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (4/4), done.  
Writing objects: 100% (6/6), 497 bytes | 0 bytes/s, done.  
Total 6 (delta 1), reused 0 (delta 0)  
To https://shwu10.cs.nthu.edu.tw/ss-student/submission-exercise.git  
  b1e0571..8a603d9  master -> master  
Branch master set up to track remote branch master from origin.  
~/SS-Projects/submission-exercise ➜ ⌂ master ➜
```

Type "git push -u origin master"

Workflow

- For each lab, you should follow the workflow below
 1. Fork our template repository on Gitlab
 2. Clone the **forked** repository to your computer
 3. Finish your lab
 4. Commit in your computer
 5. Push to Gitlab
 6. Send merge request of **your branch** to our template repository



GitLab

Go to dashboard

Project

Activity

Files

Commits

Graphs

Milestones

Issues

Merge Requests

Members

Labels

Wiki

Forks

Settings

Software Studio Student / submission-exercise ▾

Search

0 + ↗



S

submission-exercise 🔒

Forked from courses-software-studio-2017-spring

Star

0

HTTP ▾

<https://shwu10.cs.nthu.edu.tw/ss-student/>

Global ▾

1. Click Merge Requests

2 commits

109 branches

0 tags

0.17 MB

Add Changelog

Add License

Add Contribution guide

14e22e1c Finish lab1 · 4 minutes ago by Software Studio Student

Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting a lab or a project.

Try It !!

1. Fork this project.
2. Clone the project from GitLab to your local environment.
3. Add a new file and write something.
4. Commit your work.
5. Push your work to GitLab.
6. Open a merge request to the original repository.
 - Source branch: Your working branch.
 - Target branch: The branch with your student id. (e.g. [104062987](#))
 - Title: [Student ID Submission](#) (e.g. [104062987 Submission](#)).



<



GitLab

[Go to dashboard](#)

Software Studio Student / submission-exercise · Merge Requests

 Search

0 + ↗

[Open 0](#) [Merged 0](#) [Closed 0](#) [All 0](#) Filter by name ...[+ New Merge Request](#)[Author](#) [Assignee](#) [Milestone](#) [Label](#)

2. New merge request

No merge requests to show

[Project](#)[Activity](#)[Files](#)[Commits](#)[Graphs](#)[Milestones](#)[Issues 0](#)[Merge Requests 0](#)[Members](#)[Labels](#)[Wiki](#)[Forks](#)[Settings](#)

ss-student



GitLab

Go to dashboard

Project

Activity

Files

Commits

Graphs

Milestones

Issues 0

Merge Requests 0

Members

Labels

Wiki

Forks

Settings



Software Studio Student / submission-exercise > Merge Requests

New Merge Requests

Source branch

ss-student/submission-exercise

master

Finish lab1

 Software Studio Student authored 8 minutes ago

14e22e1c

Browse Files >

Target branch

courses-software-studio-2017-s...

105062558

add README

 realwei authored about 6 hours ago

b1e0571c

Browse Files >

Compare branches and continue

5. Compare branches

3. Choose the branch you pushed in your repo

4. Choose the branch named after your ID



GitLab

[Go to dashboard](#)

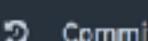
Project



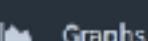
Activity



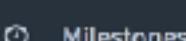
Files



Commits



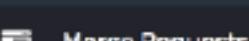
Graphs



Milestones



Issues 0



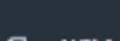
Merge Requests 0



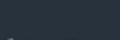
Members



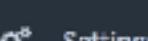
Labels



Wiki



Forks



Settings

Software Studio Student / submission-exercise · Merge Requests

 Search

① + ↗

New Merge Request

From ss-student/submission-exercise · ss-student / submission-exercise:105062558

Change branches

6. Set title to "{ID} Submission"

Title 105062558 Submission

Start the title with WIP: to prevent a Work In Progress merge request from being merged before it's ready.

Description

Write Preview

Edit in fullscreen

Markdown tip: Blocks of code can be denoted by three backticks ` `` or four leading spaces

Attach a file

Source branch

master

Target branch

105062558

Change branches

Submit merge request

Cancel

7. If everything is OK,
submit your merge request

Commits 1

Changes 1

Most recent commits displayed first

18 Feb, 2017

Finish lab1

1 commit

Software Studio Student authored 24 minutes ago

14e22e1c

Browse Files ×

Notice





gitlab



全部

圖片

影片

新聞

書籍

更多

設定

工具

約有 8,300,000 項結果 (搜尋時間 : 0.45 秒)

GitLab

<https://gitlab.com/> ▾ 翻譯這個網頁

這項網站搜尋結果說明因為網站的 robots.txt 而無法提供
瞭解詳情

GitLab.com | GitLab

<https://about.gitlab.com/gitlab-com/> ▾ 翻譯這個網頁

GitLab.com. unlimited free repositories and collaborators. Sign Up. Free public & private repositories
and unlimited collaborators. Runs GitLab Enterprise Edition ...

GitLab介紹— Practical guide for git users 0.1 文档

<git-tutorial.readthedocs.io/zh/latest/gitlab.html> ▾

GitLab介紹¶. 目前最流行的線上Git專案管理系統可以說是非GitHub 莫屬，對於一般OpenSource的專案
選擇使用GitHub做為線上Git專案管理系統即可，也免收任何 ...

GitHub - gitlabhq/gitlabhq: GitLab CE | Please open new issues in our ...

<https://github.com/gitlabhq/gitlabhq> ▾ 翻譯這個網頁

README.md. GitLab. Build status CE coverage report Code Climate Core Infrastructure Initiative Best
Practices. Canonical source. The canonical source of ...

Gitlab - 維基百科，自由的百科全書 - Wikipedia

<https://zh.wikipedia.org/zh-tw/Gitlab> ▾

GitLab是一個利用Ruby on Rails開發的開源應用程式，實現一個自代管的Git專案倉庫，可通過Web介面
進行存取公開的或者私人專案。它擁有與GitHub類似的功能， ...



GitLab.com

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](#)

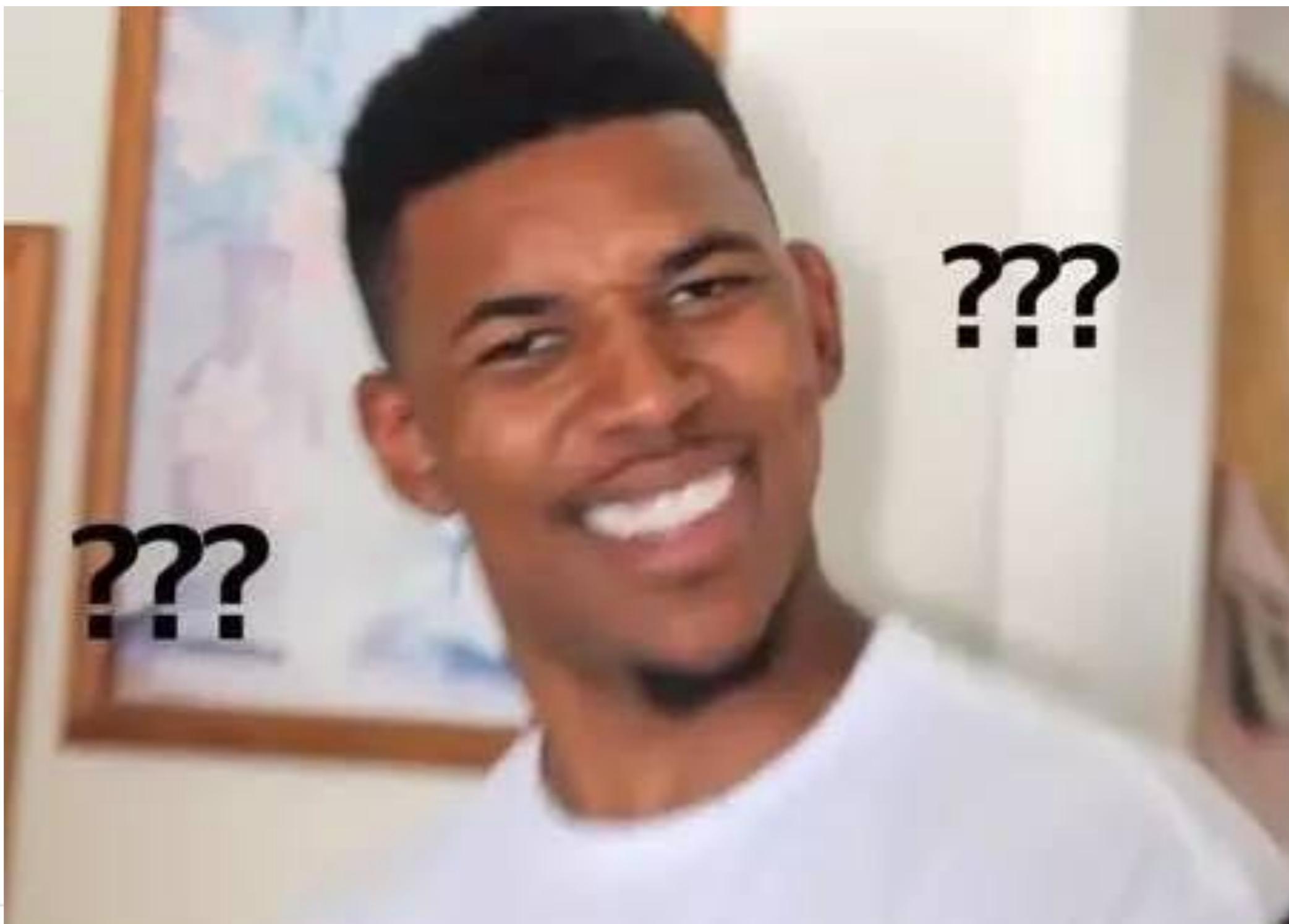
By signing up for and by signing in to this service you accept our:

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

Sign in	Register
Username or email <input type="text"/>	
Password <input type="password"/>	
<input type="checkbox"/> Remember me	Forgot your password?
Sign in	

Didn't receive a confirmation email? [Request a new one.](#)

Sign in with				
--------------	--	--	--	--



Shan-Hung Wu

Description

Announcement

Curriculum ▾

Resources

Resources

Here are some course materials and resources related to this course. For code and its details (such as assigned reading, project links, quiz, etc.) please refer to the GitLab. For online forum please refer to the iLMS system.



Here!!!!

Outline

- General Rule
- Introduction to Git
 - Version control
 - Git Basics
 - Try Git!
 - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

Tools

- Git GUI
 - GitKraken
- Editor / IDE
 - Visual Studio Code
 - Atom
 - Sublime Text
 - Brackets
 - Notepad++
 - Webstorm



axosoft

GitKraken

Repositories

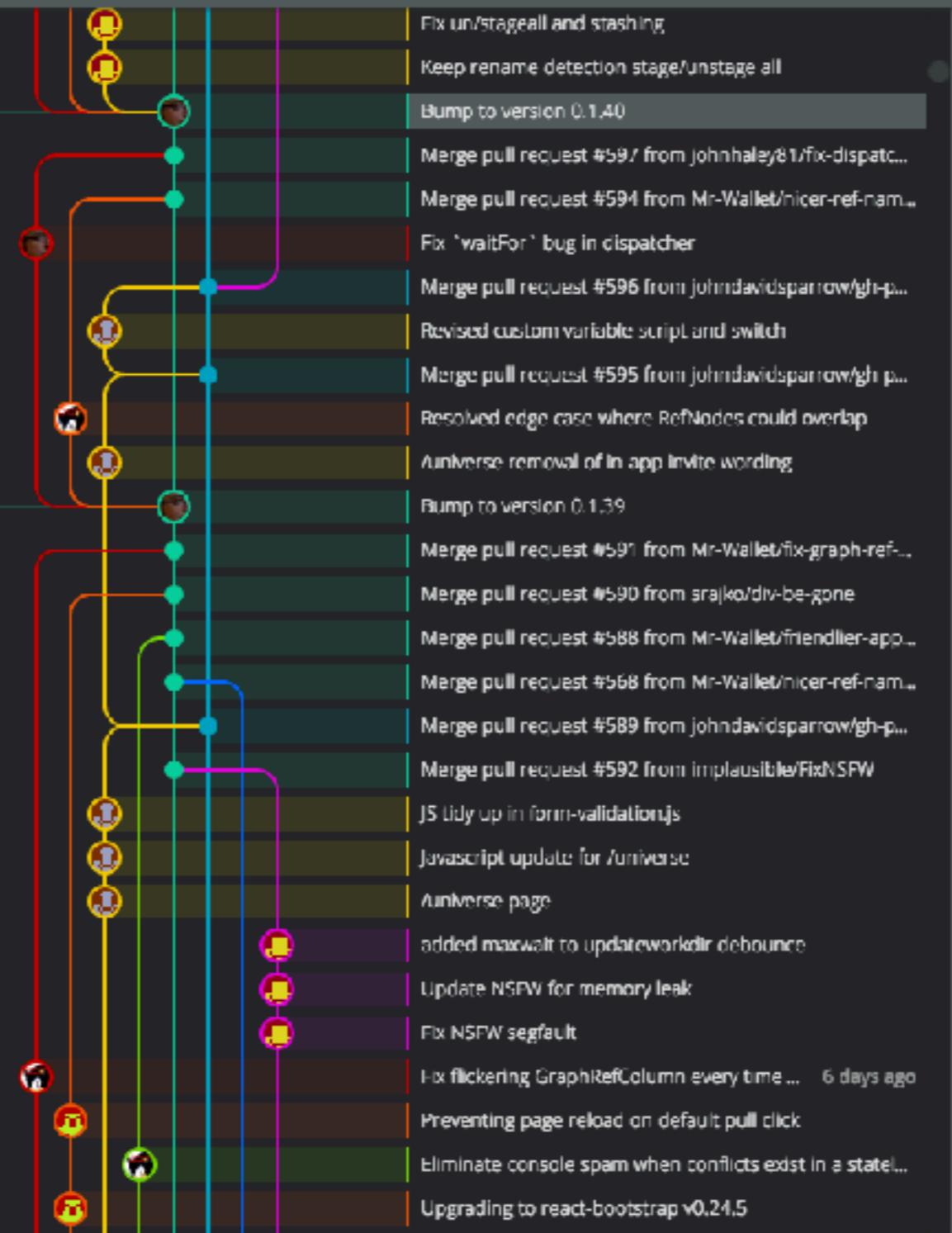
GitKraken

master



search commits

Viewing 112/151		Show All
LOCAL	(7/11)	
• folder-refbar-changes		
• fancy-responsive-refbar-n...	42/4 99+	
• graph-color-test		
• hopscotch	24/2 99+	
• init-repo-gitignore-typeahead		
• invite-system	6/2 99+	
• jars-view-file-history		
• master	5/4	
• remote-panel-redesign	15/2 13+	
• settings-theme-styling		
• view-file-history	24/2 99+	
REMOTE	(6/41)	
Jeff-Schirella	[0/0]	
Jordan Wallet	[0/7]	
Justin-GK	[0/7]	
Ken Price	[0/2]	
Kyle-Smith	[2/8]	
Max Korp	[0/2]	
Sjapan-Rajko	[0/8]	
ayreca	[0/3]	
cbargen	[0/5]	
origin	[4/4]	
TAGS	(99/99)	



This panel provides a detailed view of the commit 'Bump to version 0.1.40'. It includes the commit author (John Haley), date (September 30th 2015, 2:54 pm), and a summary of changes (0 added, 0 deleted, 2 modified). The commit message is displayed above the file differences. Two files are shown with their respective line-by-line changes:

npm-shrinkwrap.json

```
@@ -1,6 +1,6 @@  
 1 | 1 | (   
 2 | 2 | "name": "gitkraken",  
 -3 | "version": "0.1.39",  
 +3 | "version": "0.1.40",  
 4 | 4 | "dependencies": {  
 5 | 5 |   "atom-keymap": [  
 6 | 6 |     "version": "5.1.11",
```

package.json

```
@@ -1,7 +1,7 @@  
 1 | 1 | (   
 2 | 2 | "name": "gitkraken",  
 3 | "productName": "GitKraken",  
 -4 | "version": "0.1.39",  
 +4 | "version": "0.1.40",  
 5 | 5 | "description": "An intuitive git cli",  
 6 | 6 | "main": "./src/appBootstrap/main.js"  
 7 | 7 | )
```

Provide Feedback



VS Code

EXPLORER

WORKING FILES

03.jpg (img)

TBL-STYLES

css

img

js

hoverIntent.js

jquery.dropdown.js

jquery.more.js

jquery.more.min.js

jquery.plugin.js

jquery.plugin.min.js

mapper.js

maputil.js

navigation.js

smoothscroll.js

tabs.js

navigation.js

```
1 var scriptbase = _spPageContextInfo.webServerRelativeUrl + "/_layouts/15/";
2
3 $(document).ready(function () {
4     $.getScript(scriptbase + "SP.Runtime.js", function () {
5         $.getScript(scriptbase + "SP.js", function () {
6             $.getScript(scriptbase + "SP.Taxonomy.js", function () {
7                 context = SP.ClientContext.get_current();
8                 //Call your code here.
9                 console.log("Navigation - ready to rock.");
10
11                 // Get default termstore
12
13                 session = SP.Taxonomy.TaxonomySession.getTaxonomySession(context);
14                 termStore = session.getDefaultSiteCollectionTermStore();
15                 context.load(session);
16                 context.load(termStore);
17                 context.executeQueryAsync(
18                     function () {
19                         console.log('Got default term store');
20                     },
21                     function(sender, args) {
22                         console.log("Could not get default term store. " + args.get_message());
23                     }
24                 );
25
26
27             });
28         });
29     });
30 });
31
32 var topnavbar;
33
34 topnavbar += "<div class='tbl-site-navigation'>";
35 topnavbar += "    <ul class='dropdown'>";
36 topnavbar += "        <li class='''><a href='''>The Brand Code - a</a></li>";
37 topnavbar += "        <li class='dropdown1'>";
38 topnavbar += "            <ul class='sub_menu' style='visibility: hidden;'>";
39 topnavbar += "                <li class='large'>";
40 topnavbar += "                    <div class='dropdownbox'>";
41 topnavbar += "                        <div class='dropdownbox-title'>Welcome to the Brand Code</div>";
42 topnavbar += "                        <ol>";
43 topnavbar += "                            <li><a href='''>The Importance of Brand Building</a></li>";
44 topnavbar += "                            <li><a href='''>Introduction to the Brand Code</a></li>";
45 topnavbar += "                            <li><a href='''>You and the Brand Code</a></li>";
46 topnavbar += "                        </ol>";
47 topnavbar += "                    </div>";

```

Ln 38, Col 72 UTF-8 CRLF JavaScript



ATOM

A hackable text editor for the 21st Century

The screenshot shows the Atom code editor interface. On the left, there's a sidebar with a tree view of project files: build, docs, dot-atom, exports, keymaps, menus, node_modules, resources, script, spec, src (which is selected), and static. The main editor area displays the contents of the atom.coffee file. The file contains several lines of code, including comments and a class definition. The code editor has a dark theme with syntax highlighting for different programming constructs.

```
atom.coffee • Settings
```

```
18
19 # Essential: Atom global for dealing with packages, themes, menus, and the window system.
20 #
21 # An instance of this class is always available as the `atom` global.
22 module.exports =
23   class Atom extends Model
24     @version: 1 # Increment this when the serialization format changes
25
26     # Load or create the Atom environment in the given mode.
27     #
28     # Returns an Atom instance, fully initialized.
29     @loadOrCreate: (mode) ->
30       startTime = Date.now()
```

Reference

- Learn Git branching (interactive)
 - <http://pcottle.github.io/learnGitBranching/>
- Pro Git
 - <http://git-scm.com/book/>
- 寫給大家的 Git 教學
 - <http://www.slideshare.net/littlebtc/git-5528339>

Today's exercise

- Install Git command line tool in your computer.
 - Follow appendix A.
- Try to submit in GitLab.
 - No score but very important!