Lab 01
How to Survive & Introduction to Git

Software Studio
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

Cheng-Wei Tseng
曾振瑋

Tz-Yu Lin
林子瑜

Rosalie J. Dolor
羅莎莉

Victor R. Sanchez
桑維克

How to Find Us?

- Office Hour (TAs)
  - Thu. 1:00-3:00pm at Delta 723/724
- Email (**Personal question only**)
  - Cheng-Wei Tseng : cwtseng@datalab.cs.nthu.edu.tw
  - Tz-Yu Lin : tylin@datalab.cs.nthu.edu.tw
  - Rosalie J. Dolor : rdolor@datalab.cs.nthu.edu.tw
  - Victor R. Sanchez : vrsanchez@datalab.cs.nthu.edu.tw
- Online Forum
  - iLms

The Policy of Labs

- **All labs need to be submitted to GitLab.**
- Late submission will **not** be accepted.
- Plagiarism will not be tolerated.
  - If we find you copy someone's code, you will get **0 point** for that lab.
- Grading
  - Submission before lab ends gets 100% score
  - Submission before **11:59pm** gets 60% score

Grading Example

- 4 problems, 25% each
- Solved 4 during the lab
  - 100
- Solved 3 during the lab, 1 before 11:59pm
  - 75 + 25*0.6 = 90
- Solved 4 after the lab, before 11:59pm
  - 100 * 0.6 = 60

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 **2/24 23:59**
    - Register form : https://goo.gl/vlJCZX
    - After that day we will match the rest student.
    - Remember the Landing Page Demo is on 3/14.

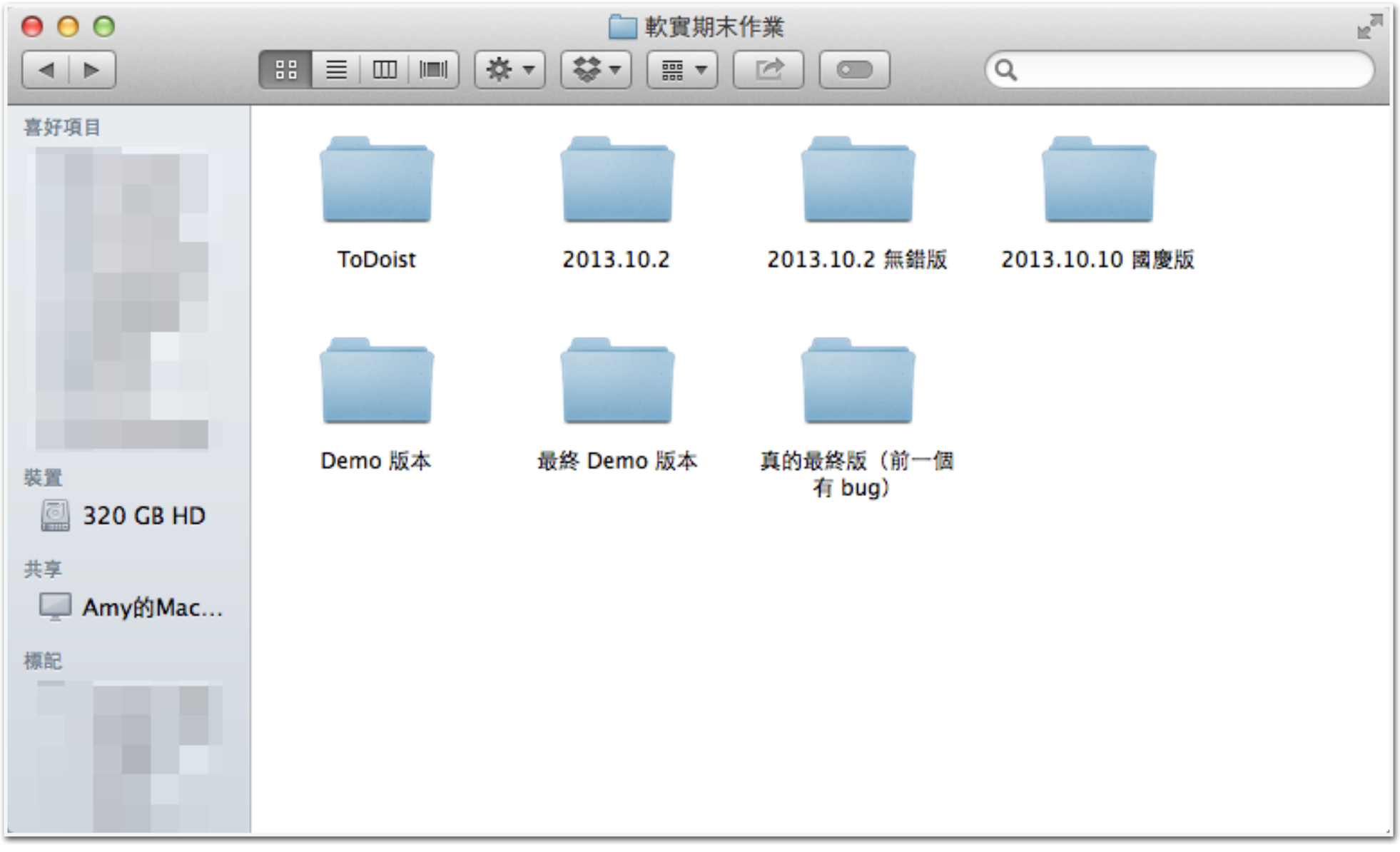
Outline

- 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



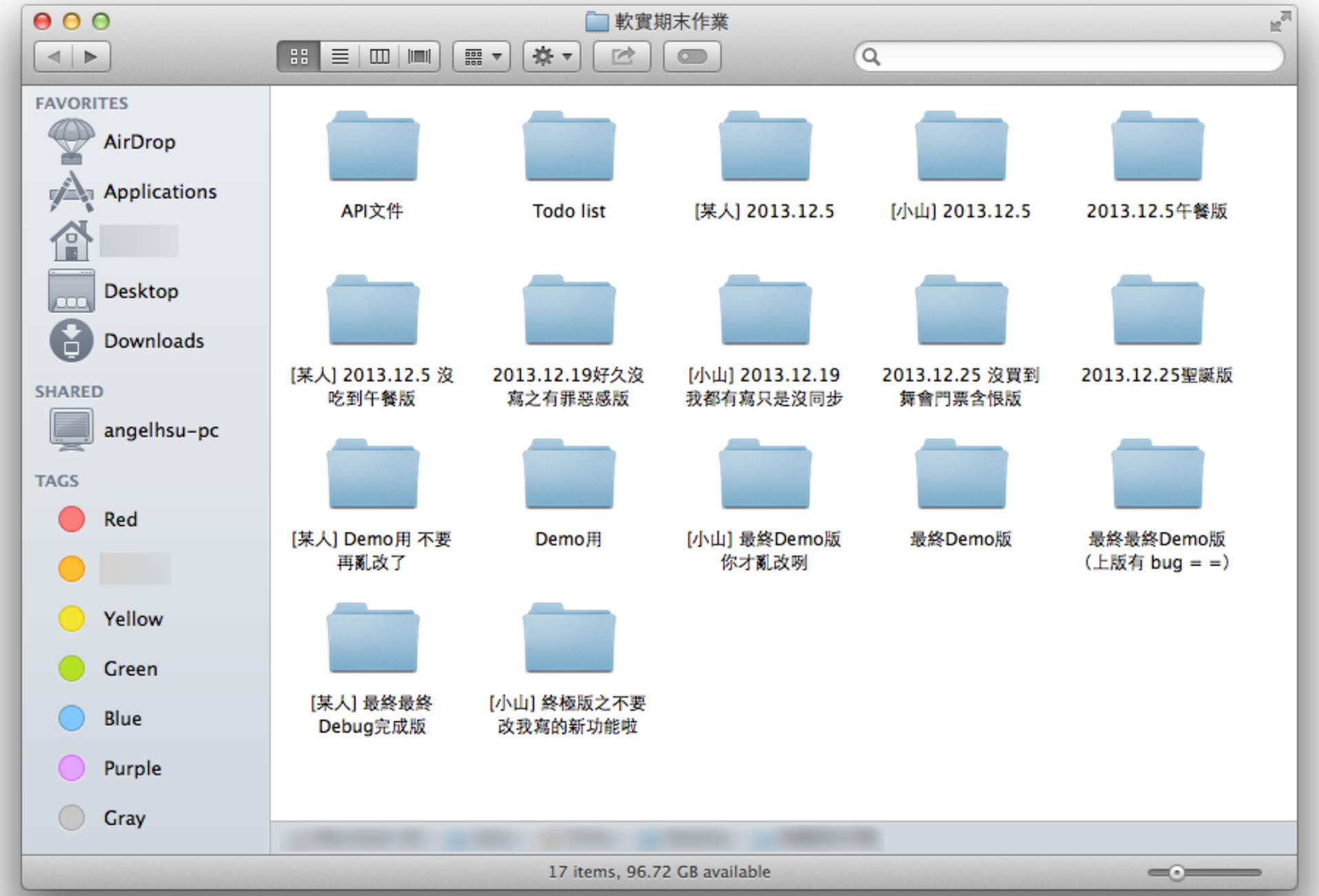
How to work with others?

?

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

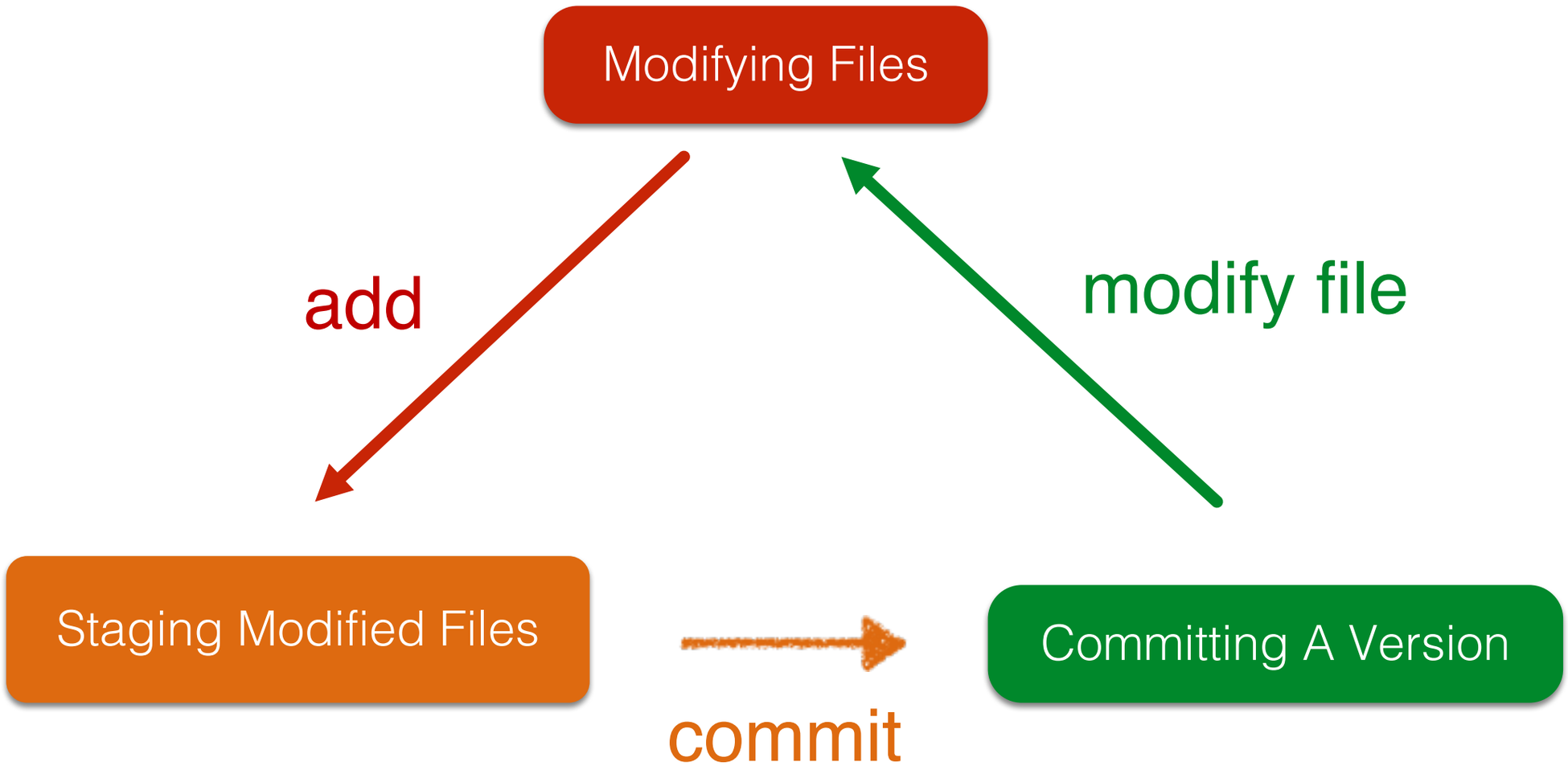
- Git is a popular version control system which is
  - Fast
  - Easy to use
  - Distributed
- A git repository is a mini database that tracks your files.

Git Workflow (1/2)

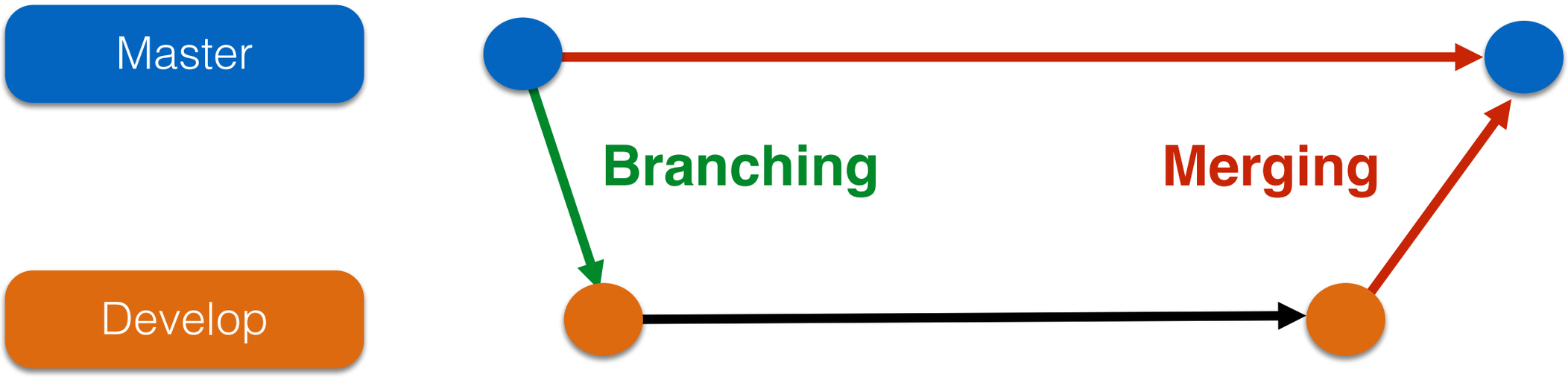
- With a local repository in your computer, you'll need following operations to make git track your work:

1. Create/modify files
2. Let git monitor the files by *adding* them to staging files.
3. *Commit* your changes to and git will create snapshots (versions) of the files for you.

Git Workflow (2/2)



Git Branch



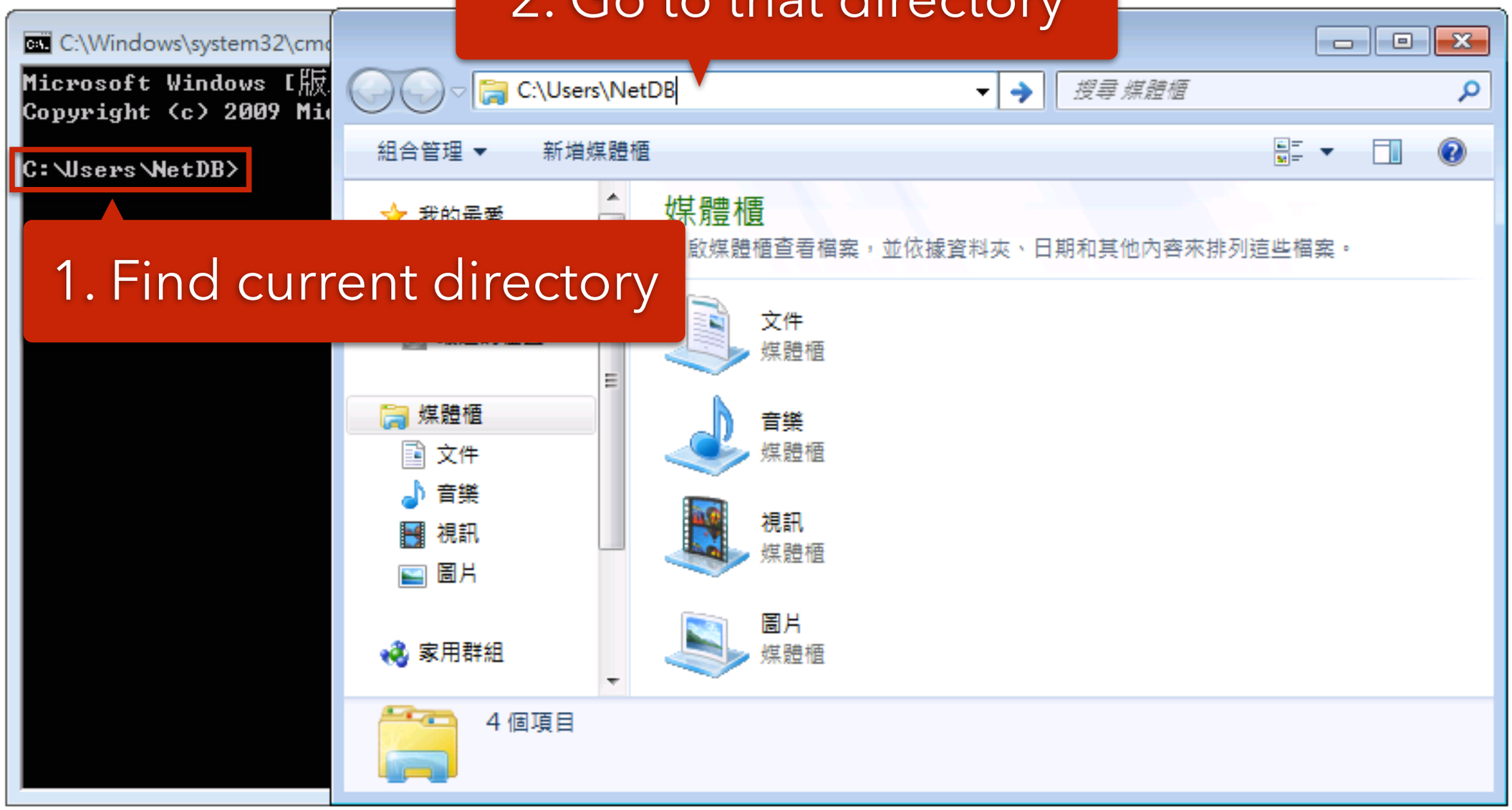
Outline

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

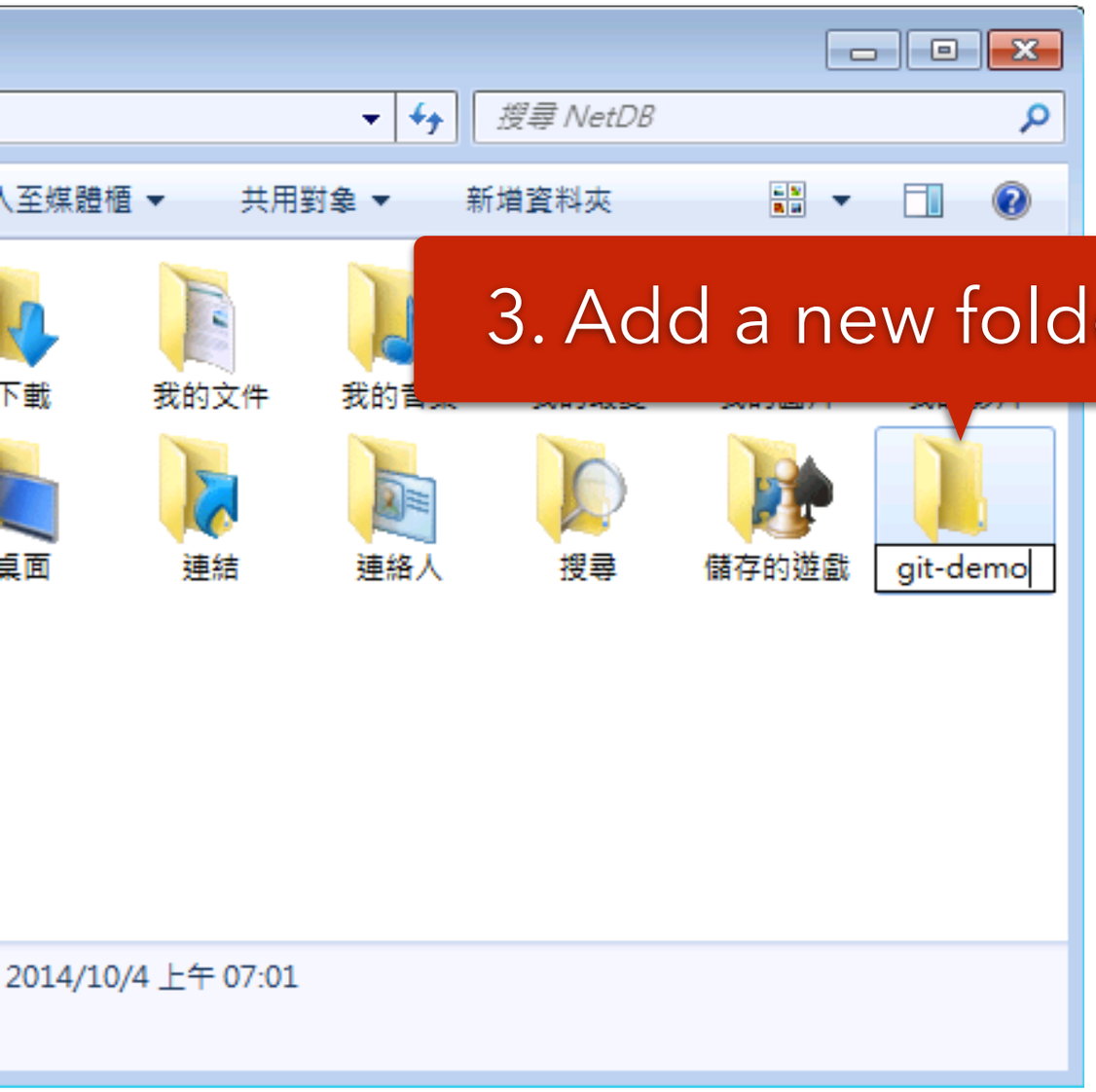
Be Professional

Basic Git Commands (1/2)

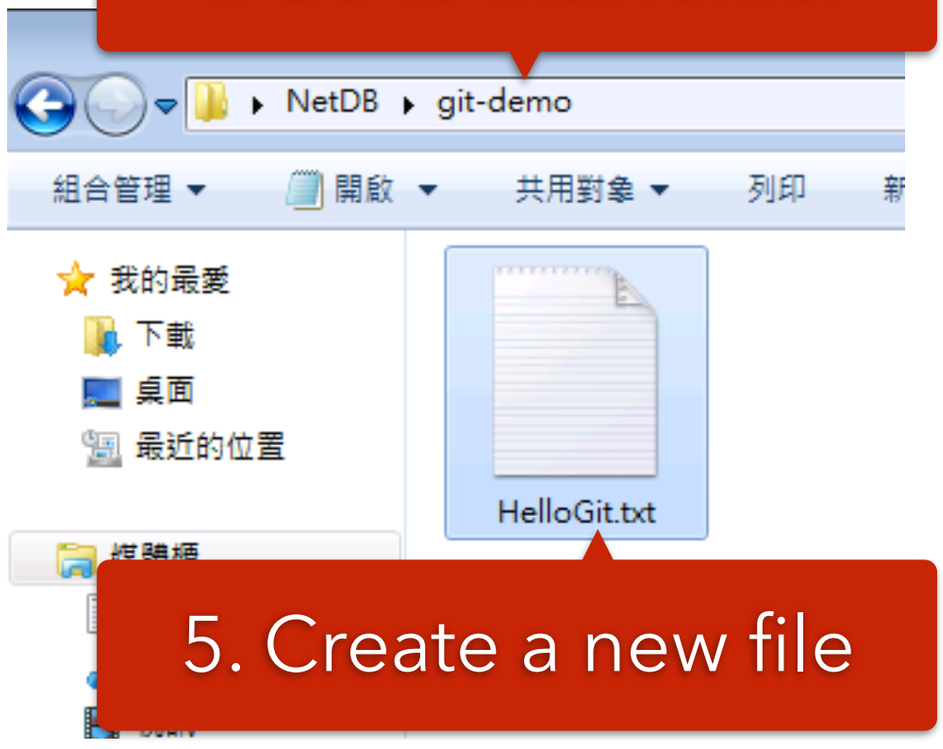
- **`git init`**
  - Initialize a repository at current directory.
- **`git add [file_name]`**
  - Add files to git repository and let git track them.
- **`git commit –m "commit messages"`**
  - Save the changes to the git repository and create snapshots of the files.
- **`git checkout [version]`**
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


2. Go to that directory
C:\Windows\system32\cmd
Microsoft Windows [版
Copyright (c) 2009 Mi
C:\Users\NetDB>
1. Find current directory
C:\Users\NetDB
搜尋 媒體櫃
組合管理
新增媒體櫃
媒體櫃
啟媒體櫃查看檔案，並依據資料夾、日期和其他內容來排列這些檔案。
文件
媒體櫃
音樂
媒體櫃
視訊
媒體櫃
圖片
媒體櫃
媒體櫃
文件
音樂
視訊
圖片
家用群組
4 個項目

X

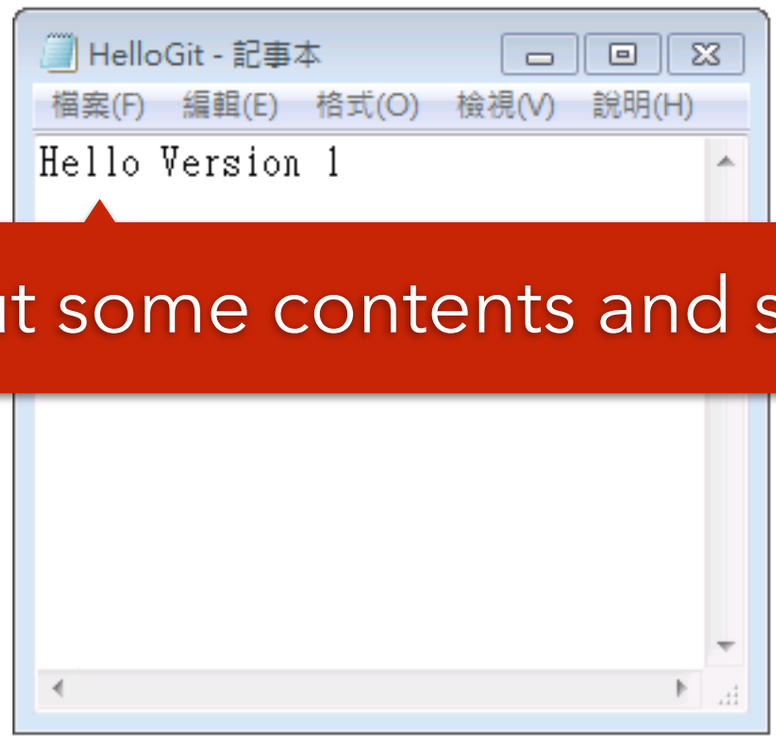


3. Add a new folder



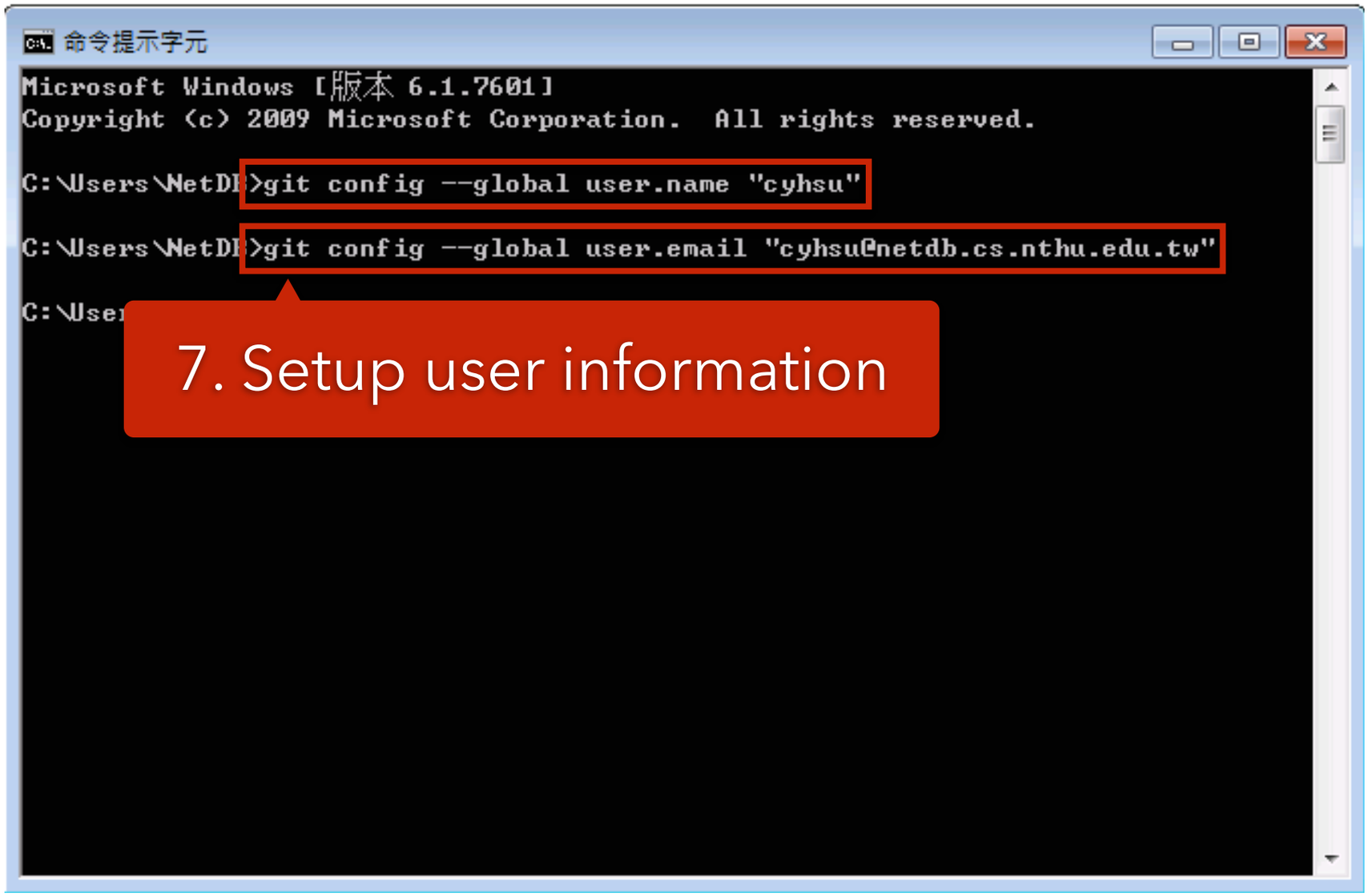
4. Go to that folder

5. Create a new file

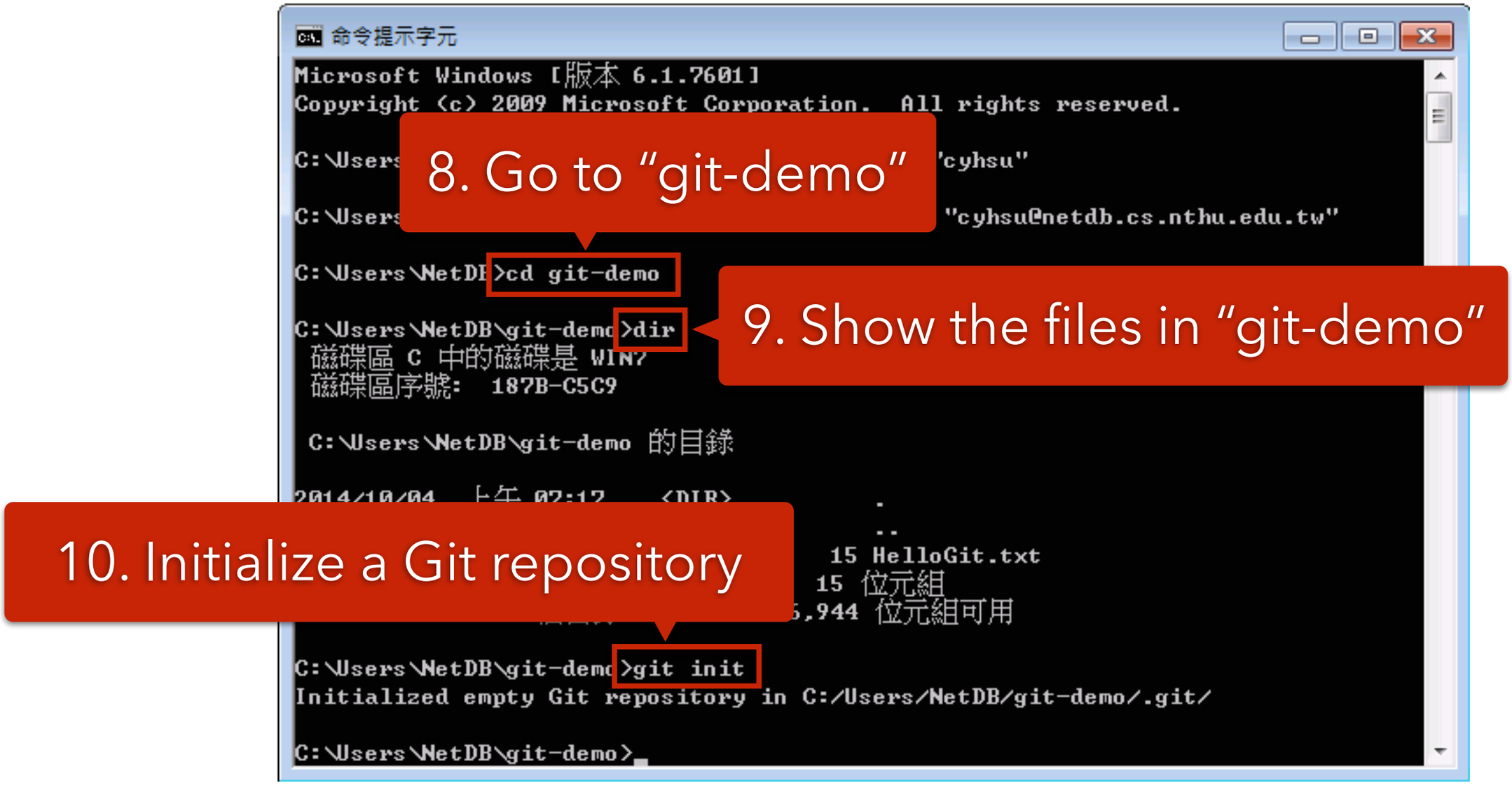


6. Put some contents and save

X

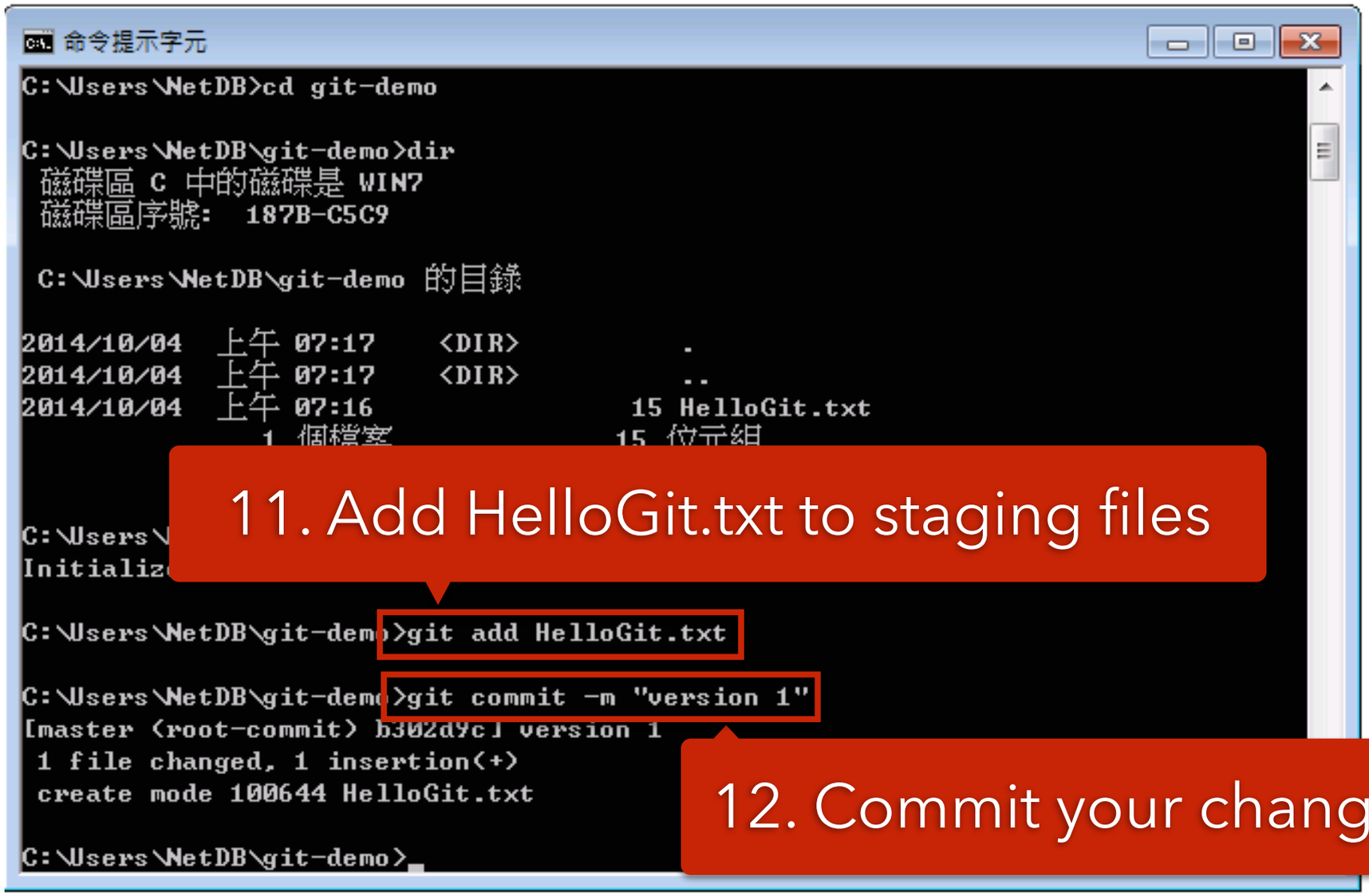



```
$ git config --global user.name "name"
$ git config --global user.email "email"
```




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

X

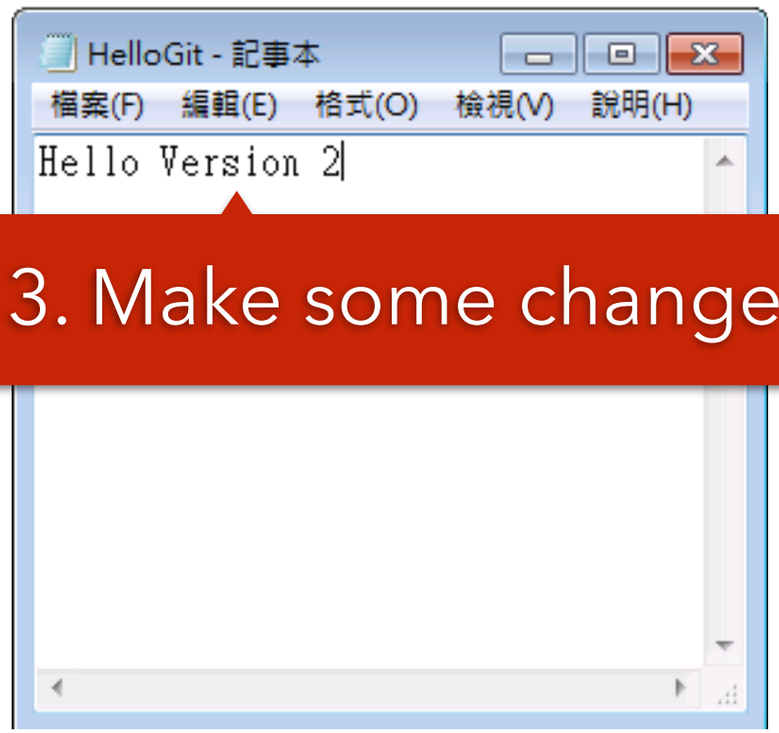


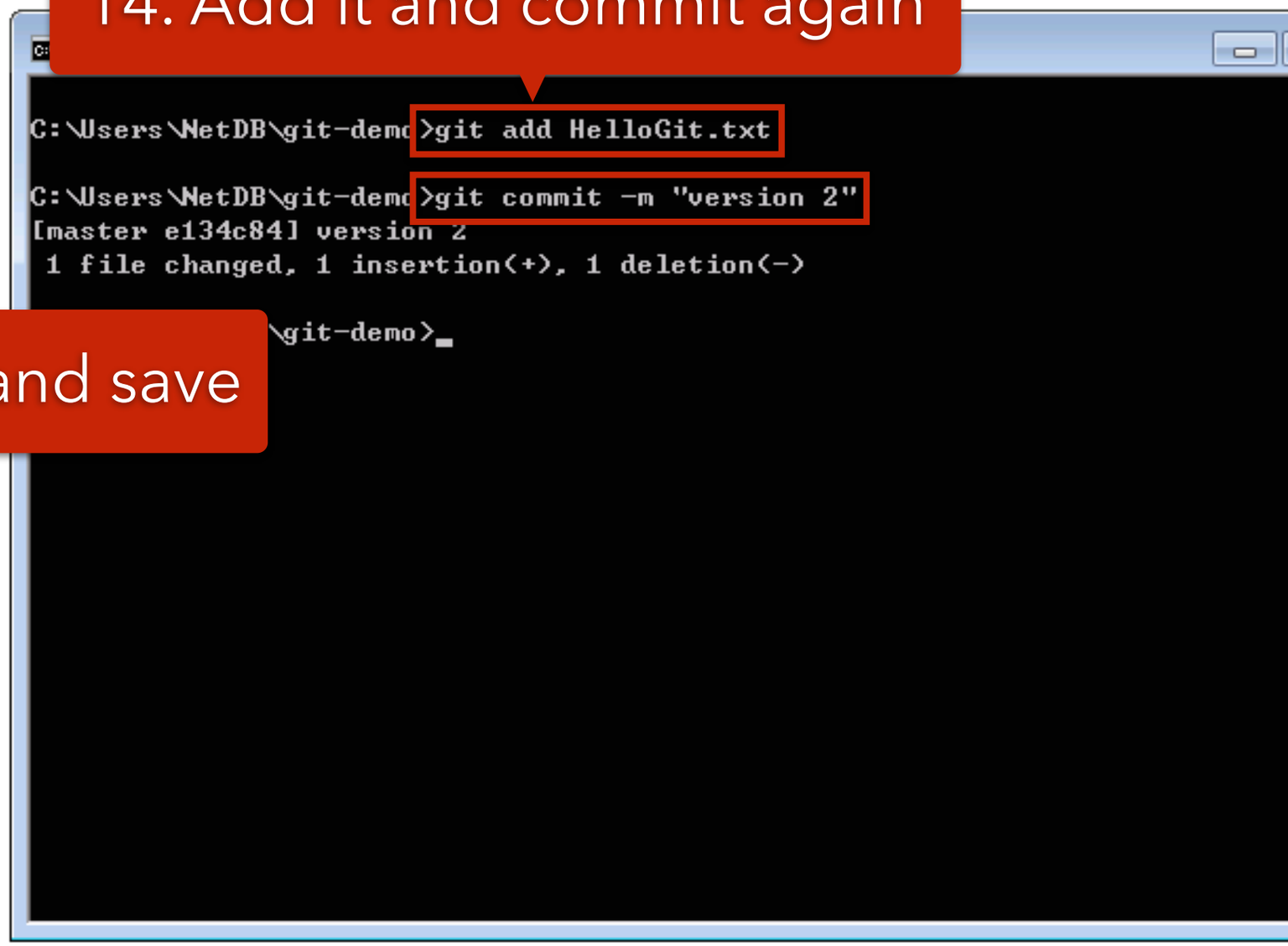

```
# Add HelloGit.txt to staging files
$ git add HelloGit.txt

# Commit the changes to the repository
# where "version 1" is the commit message
$ git commit -m "version 1"
```

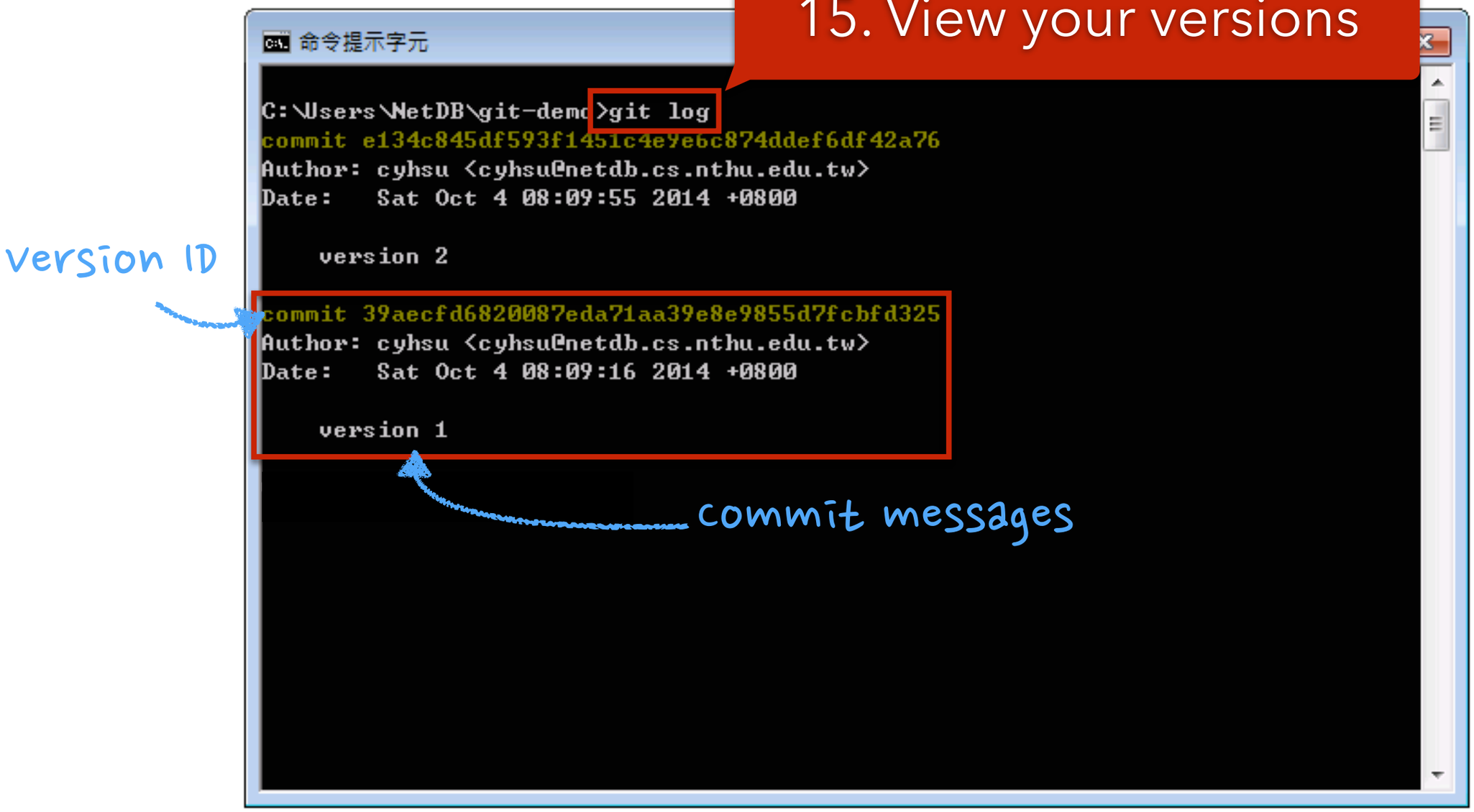
14. Add it and commit again

13. Make some changes and save

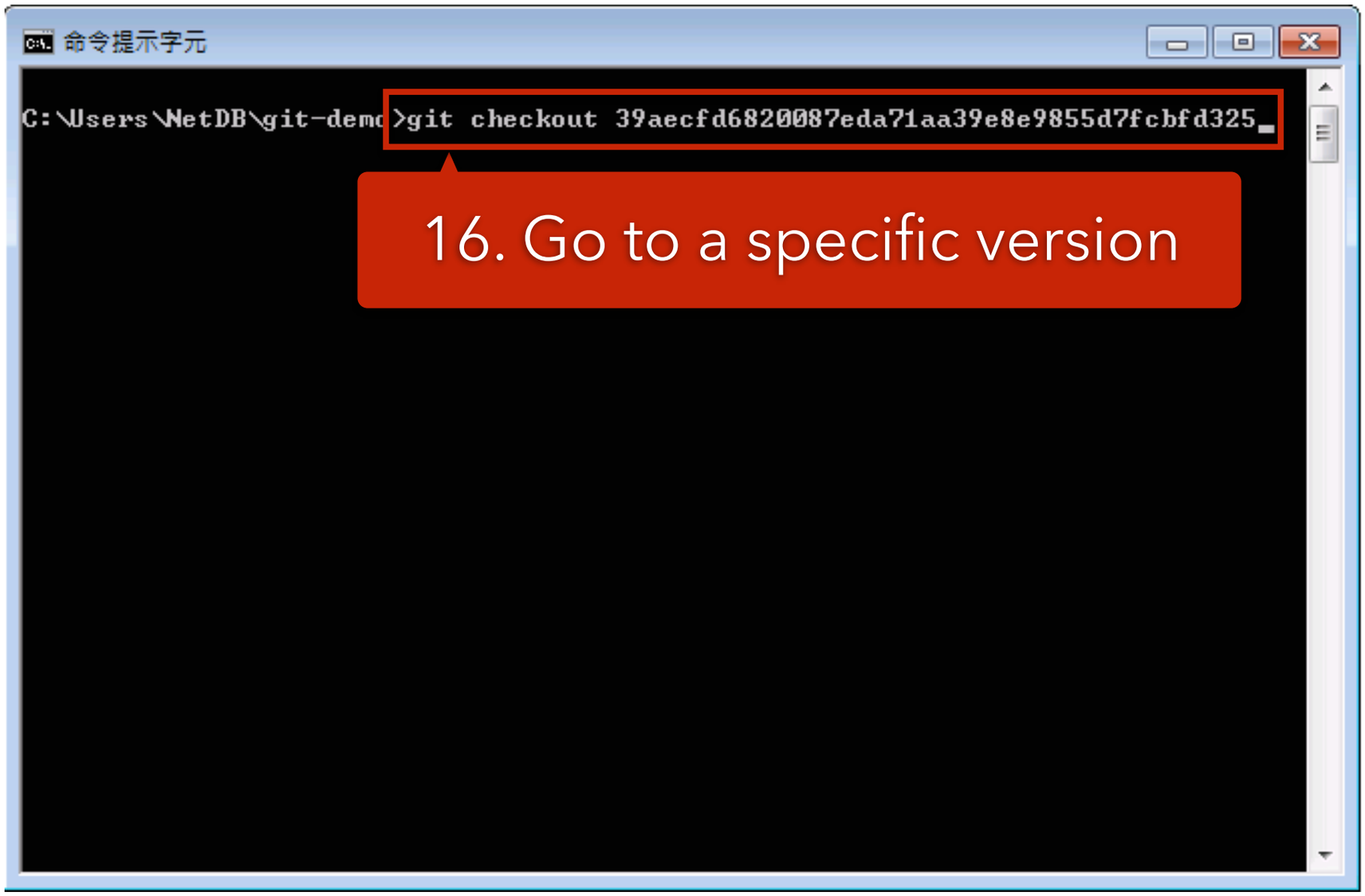




15. View your versions




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




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

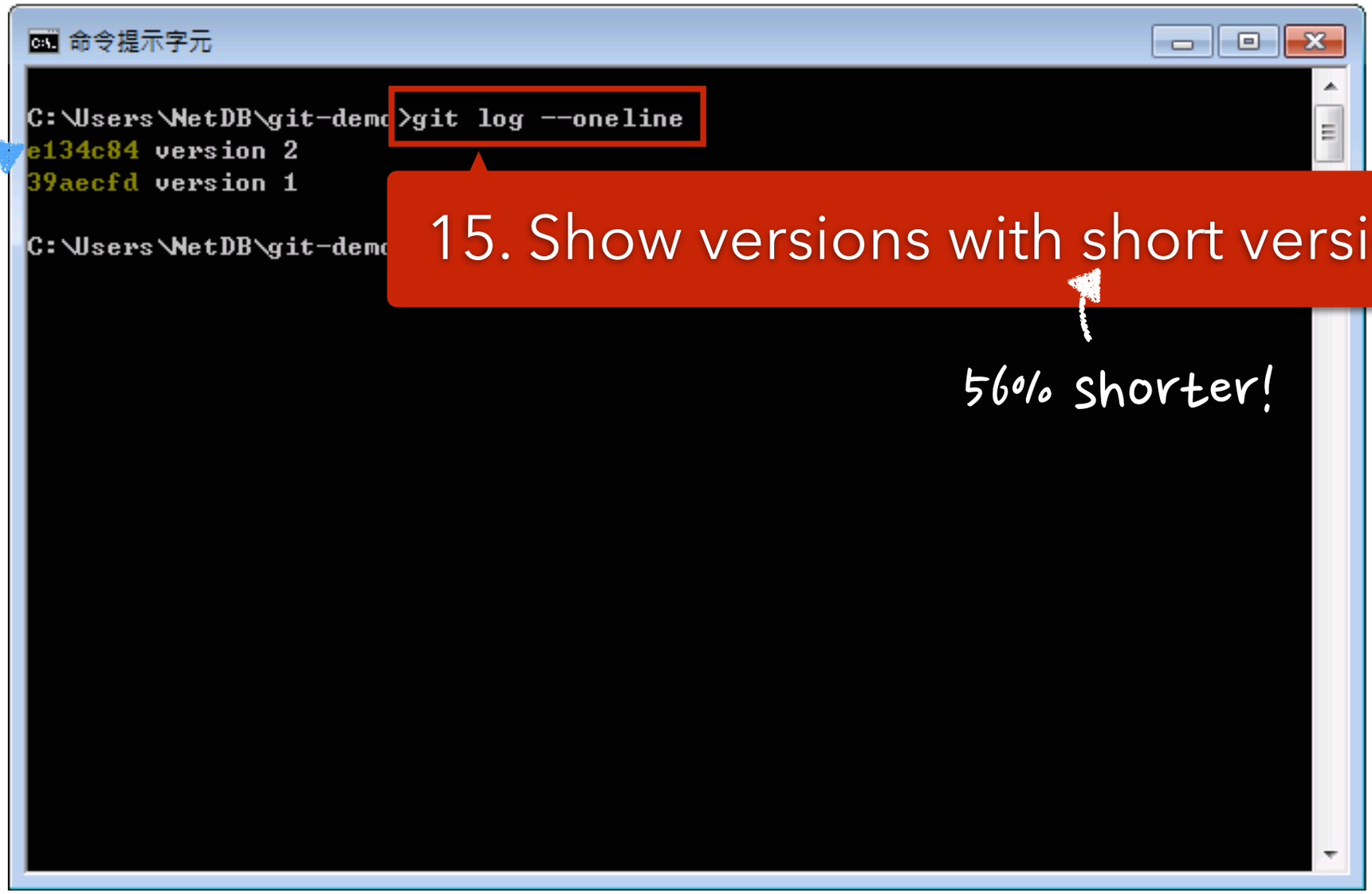
X

LIFE IS TOO SHORT TO TYPE THAT VERSION ID!

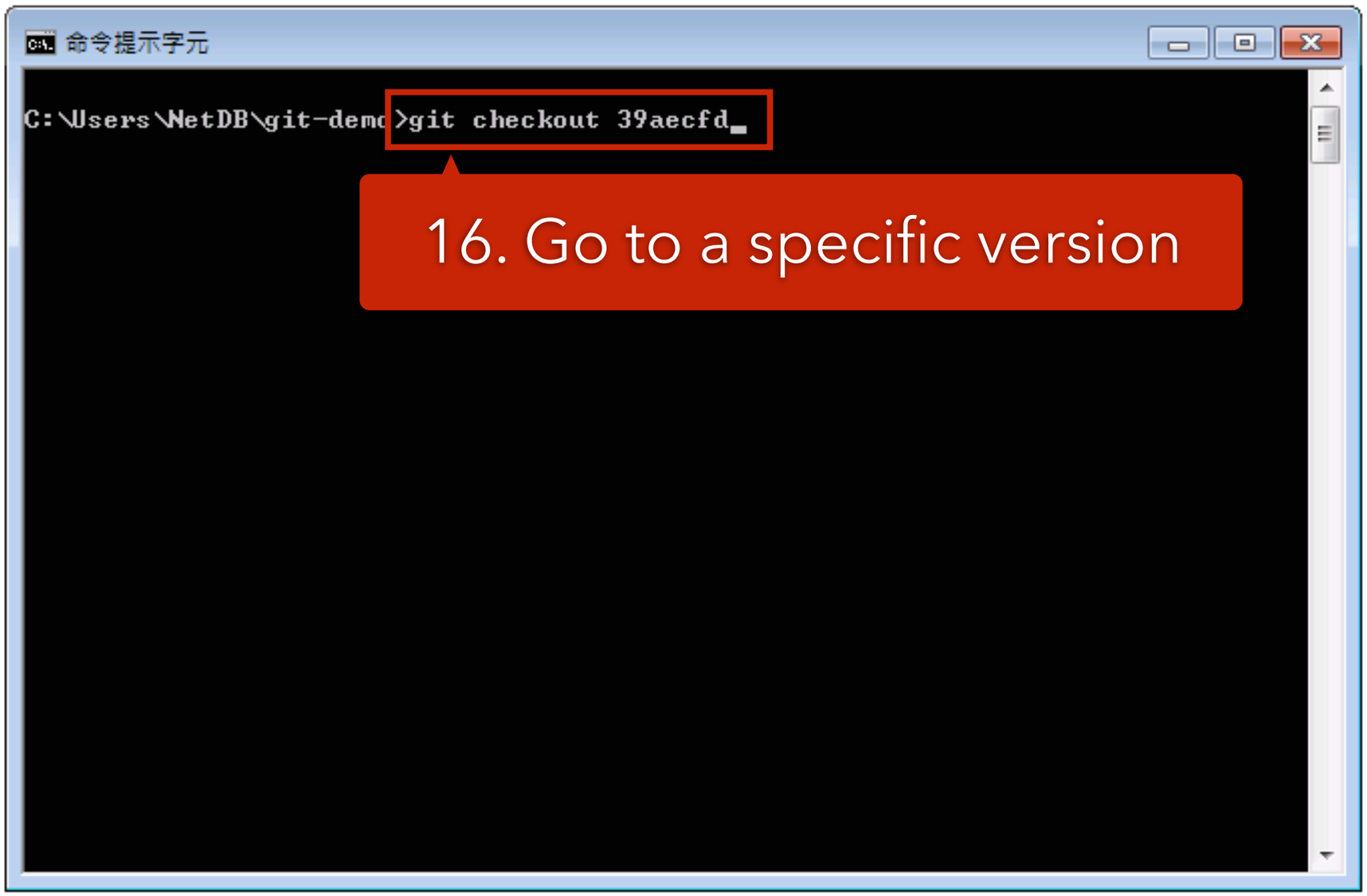
which is 40 characters long…

X

version ID

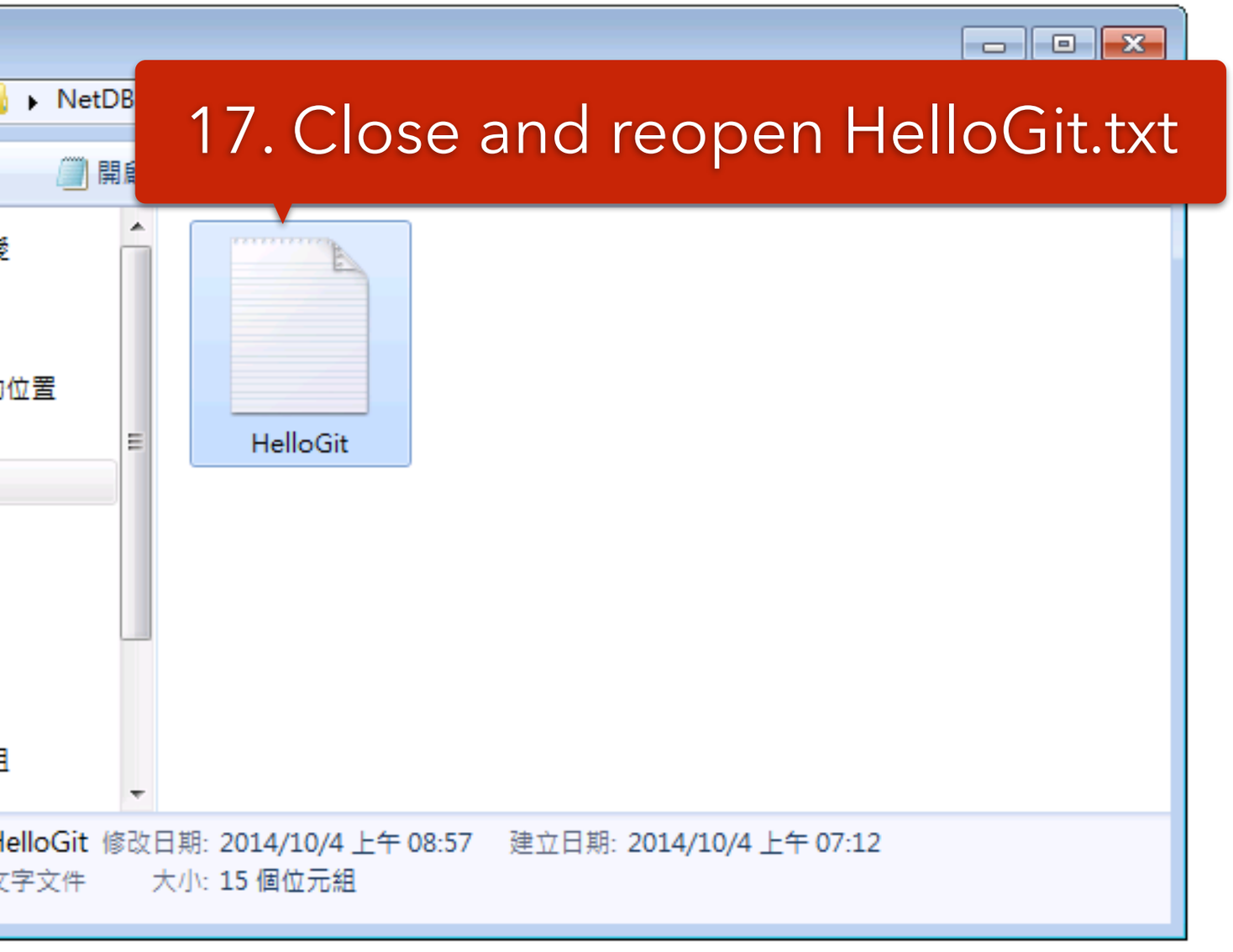


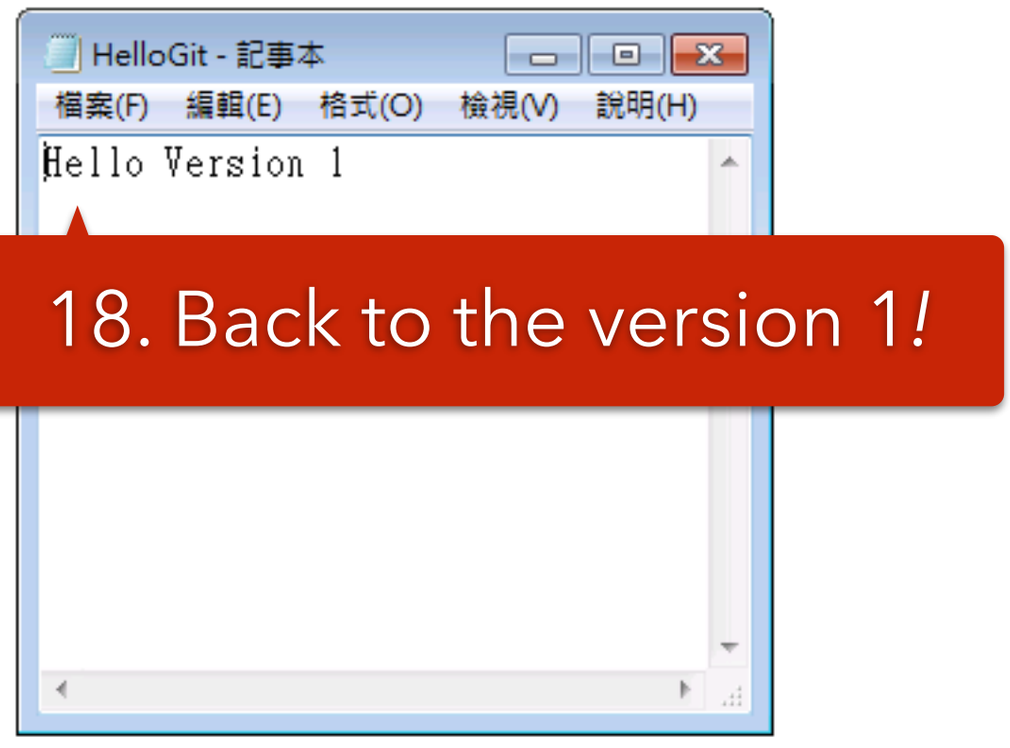

```
# Show versions with short version id
$ git log --oneline
```




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

X


17. Close and reopen HelloGit.txt
HelloGit
HelloGit 修改日期: 2014/10/4 上午 08:57 建立日期: 2014/10/4 上午 07:12
大小: 15 個位元組


HelloGit - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Hello Version 1
18. Back to the version 1!

X

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)

Server

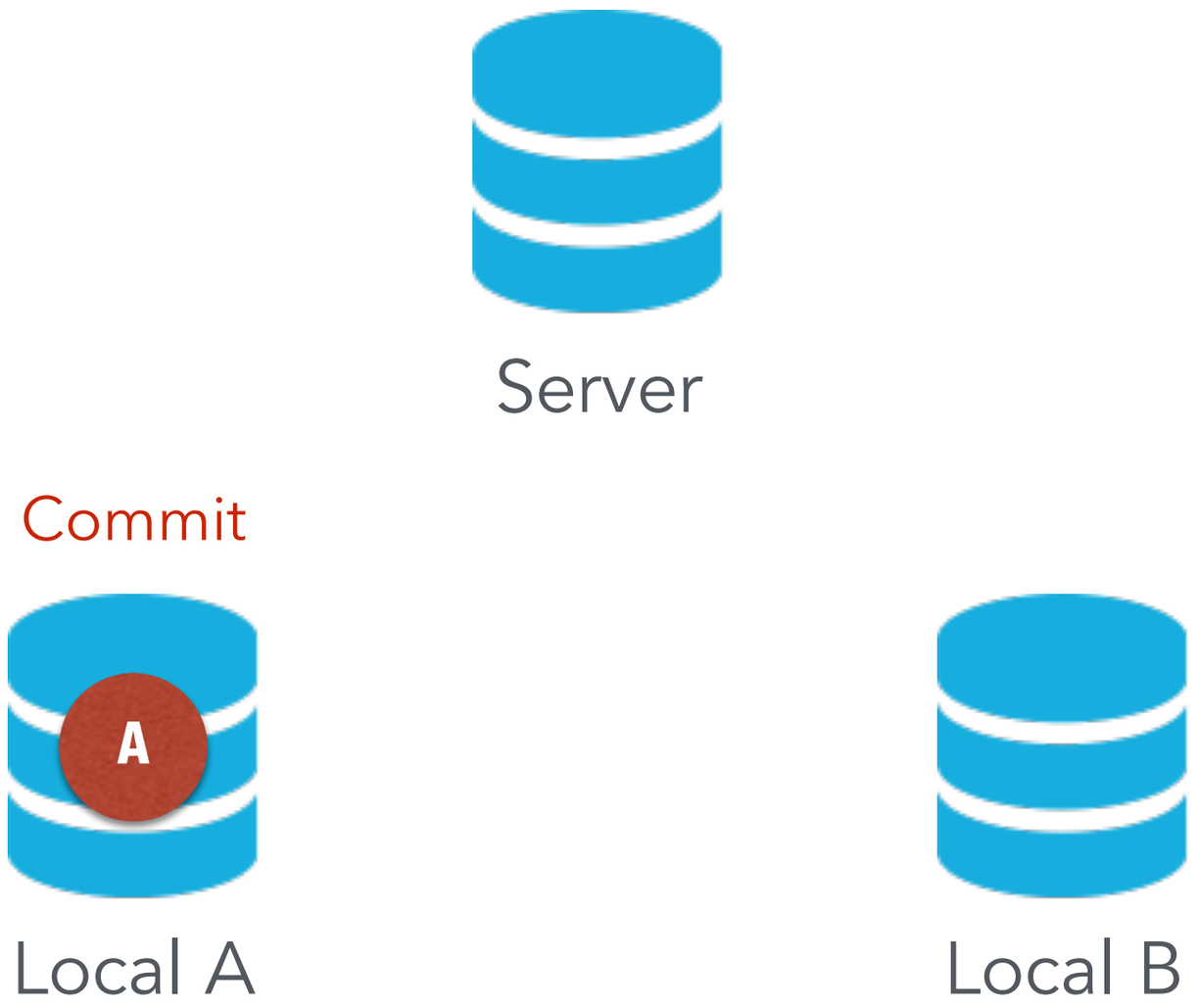
Clone

Local A

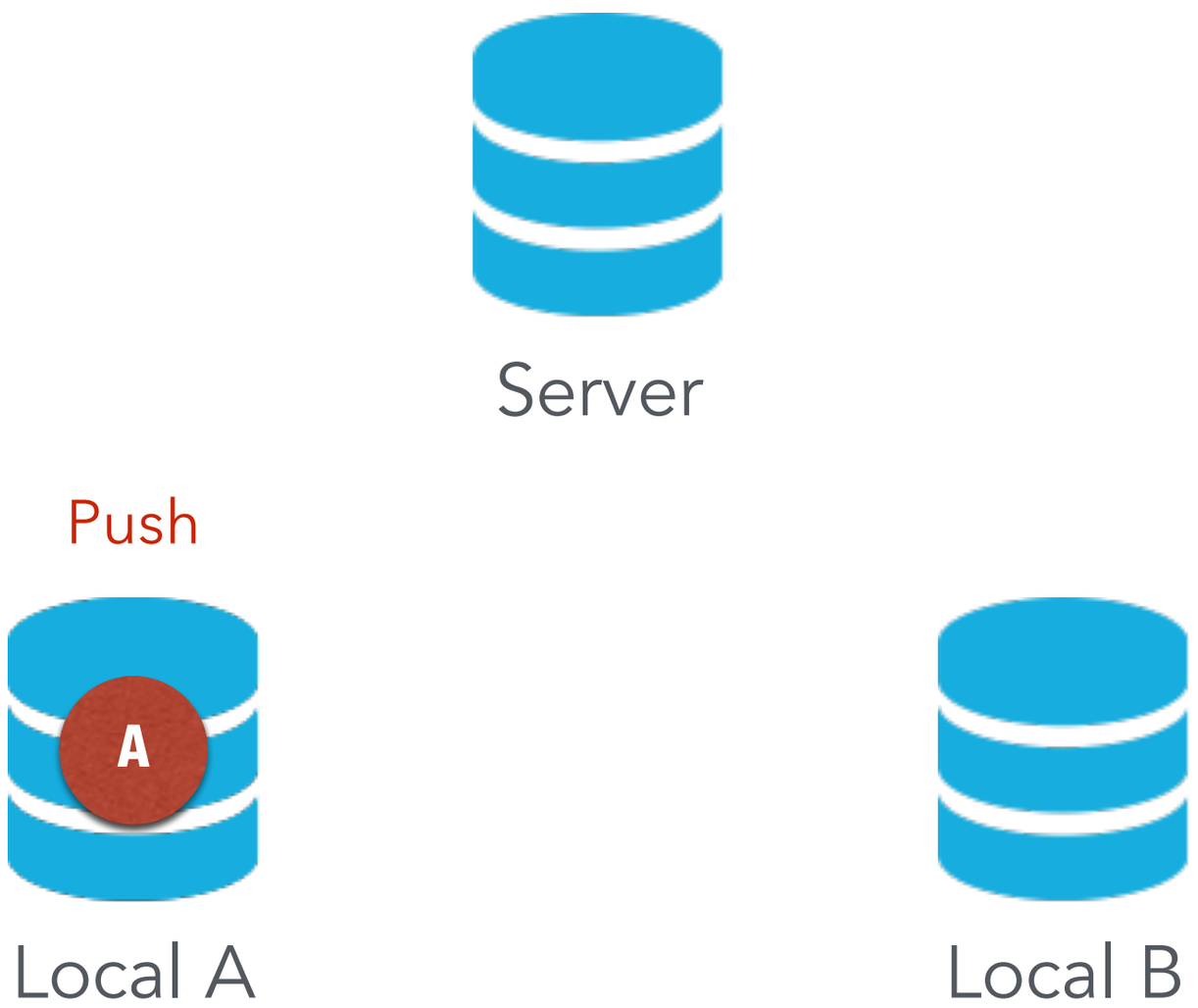
Clone

Local B

Collaboration Workflows (2/2)



Collaboration Workflows (2/2)



Collaboration Workflows (2/2)

Server

Pull

Local A

Local B

Collaboration Workflows (2/2)

Server

Pull

Local A

Local B

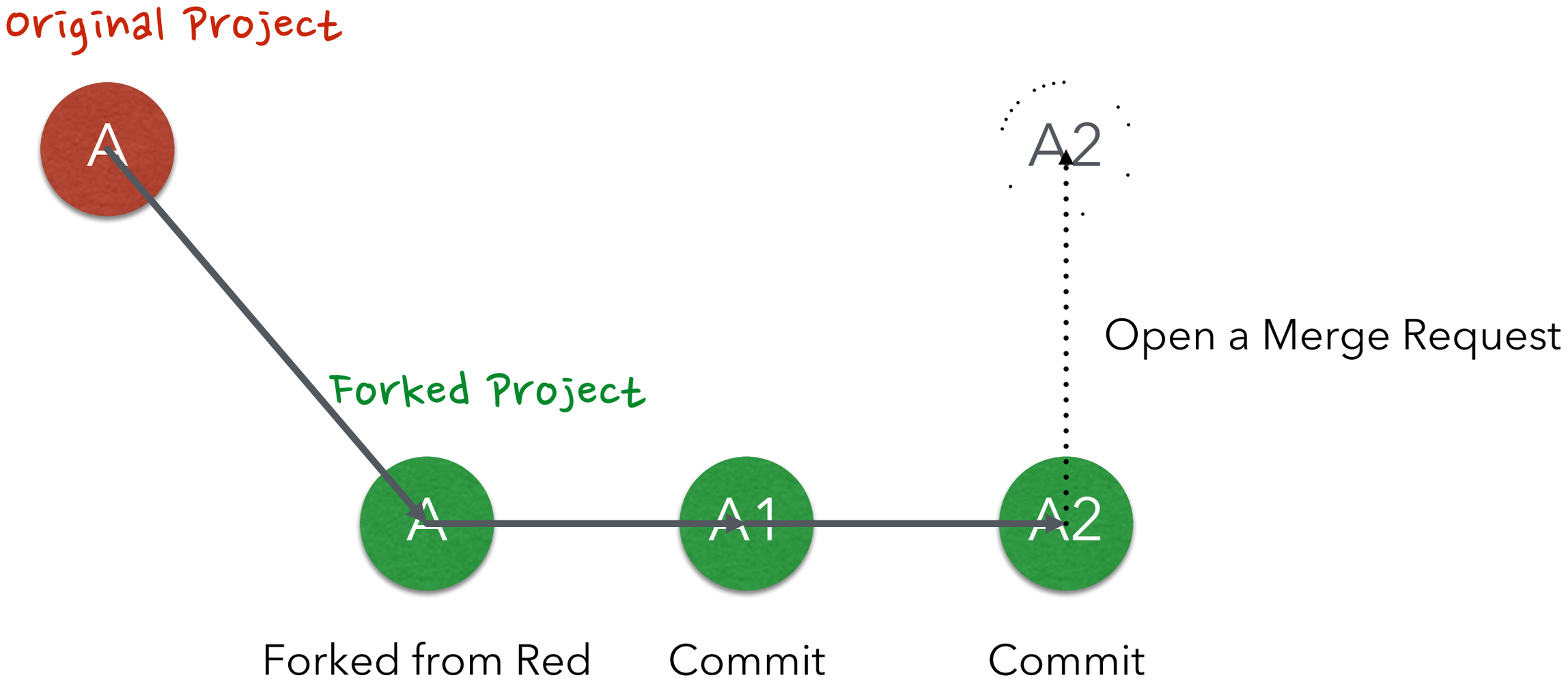
Why Authentication Failed?

Collaboration Workflow

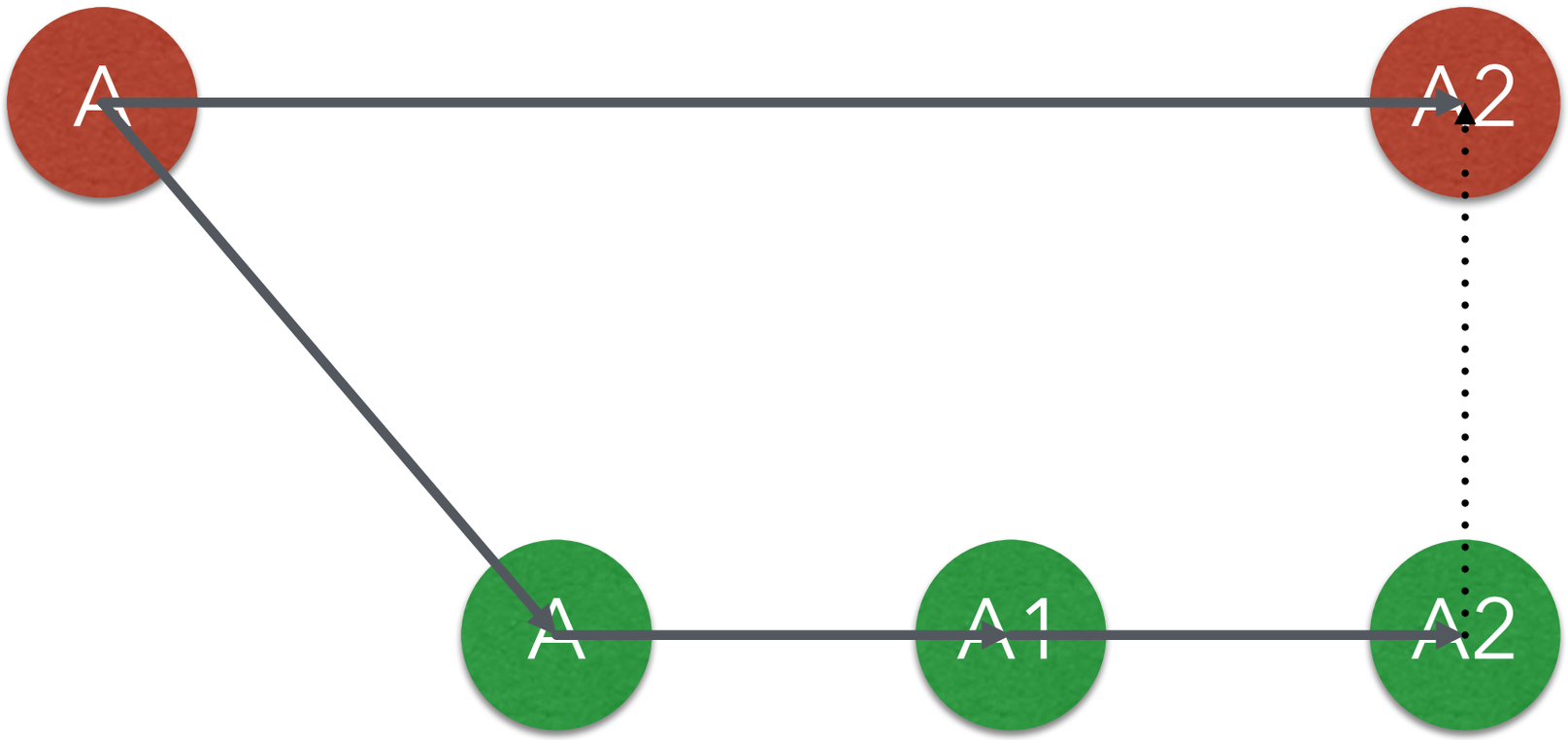
- If you tried to clone the code template from a server and want to push the modified file.
    - You will get authentication failed.
        - It’s because it was a project of others, which means you are not able to save the changes back to the server.
- So, how can I copy a project from others on a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

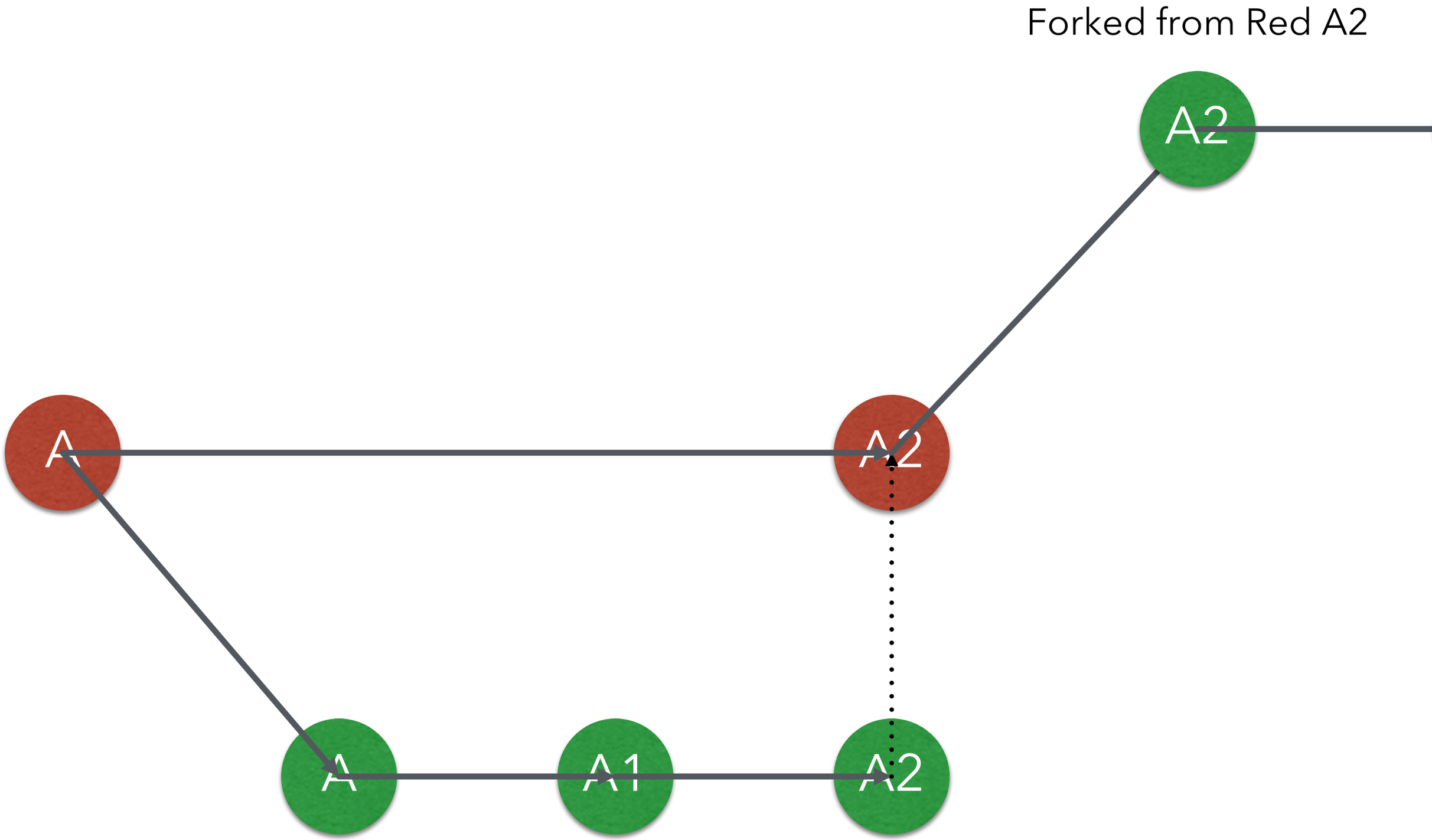
Introducing

Fork


Original Project
A
Forked Project
A
A1
A2
A2
Open a Merge Request
Forked from Red
Commit
Commit


A
A
A1
A2
A2
✔ Accept


A
A2
A
A1
A2


Forked from Red A2
A2
A
A2
A
A1
A2

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

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

courses-software-studio-2017-spring

Search

0

Go to dashboard

Group

Activity

Milestones

Issues 0

Merge Requests 0

Members

courses-software-studio-2017-spring

@courses-software-studio-2017-spring

All Projects

Filter by name

Last updated

S submission-exercise

ss-student

Go to group

Project
Activity
Files
Commits
Graphs
Milestones
Issues 0
Merge Requests 0
Labels
Wiki
Forks

S

submission-exercise

☆ Star 0 | Fork 0

1. Click to fork

HTTP https://shwu10.cs.nthu.edu.tw/ Global

1 commit | 109 branches | 0 tags | 0.48 MB

b1e0571c add README · about 4 hours ago by realwei

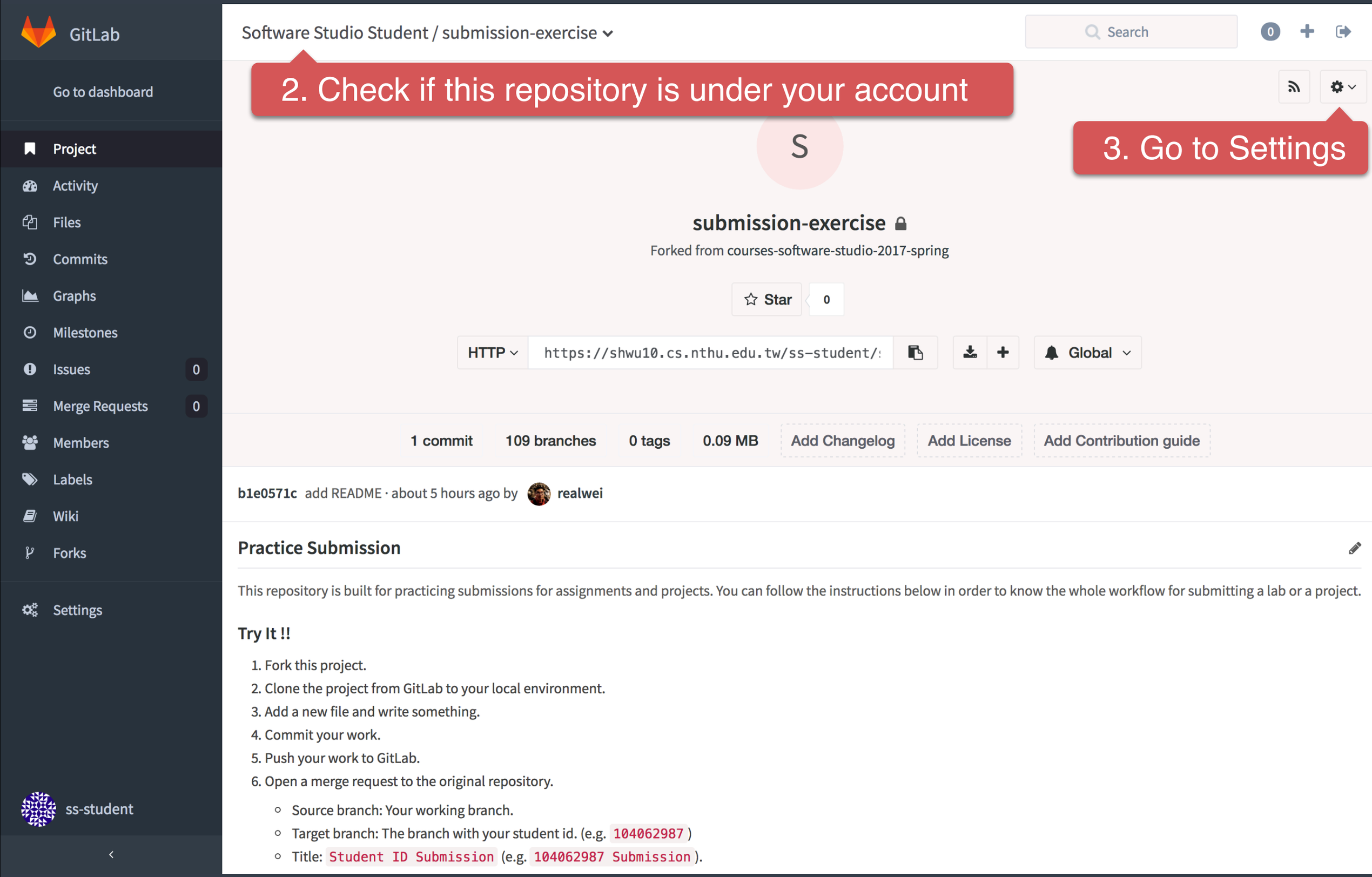
Practice Submission

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
ss-student
Software Studio Student / submission-exercise
Search
2. Check if this repository is under your account
3. Go to Settings
S
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).

GitLab

Go to project

- Project Settings
- Groups
- Deploy Keys
- Webhooks
- Services
- Protected Branches

ss-student

Software Studio Student / submission-exercise · Settings

Search

0

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

Software Studio Student / submission-exercise · Settings

Search

0

Go to project

Project Settings

Groups

Deploy Keys

Webhooks

Services

Protected Branches

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
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

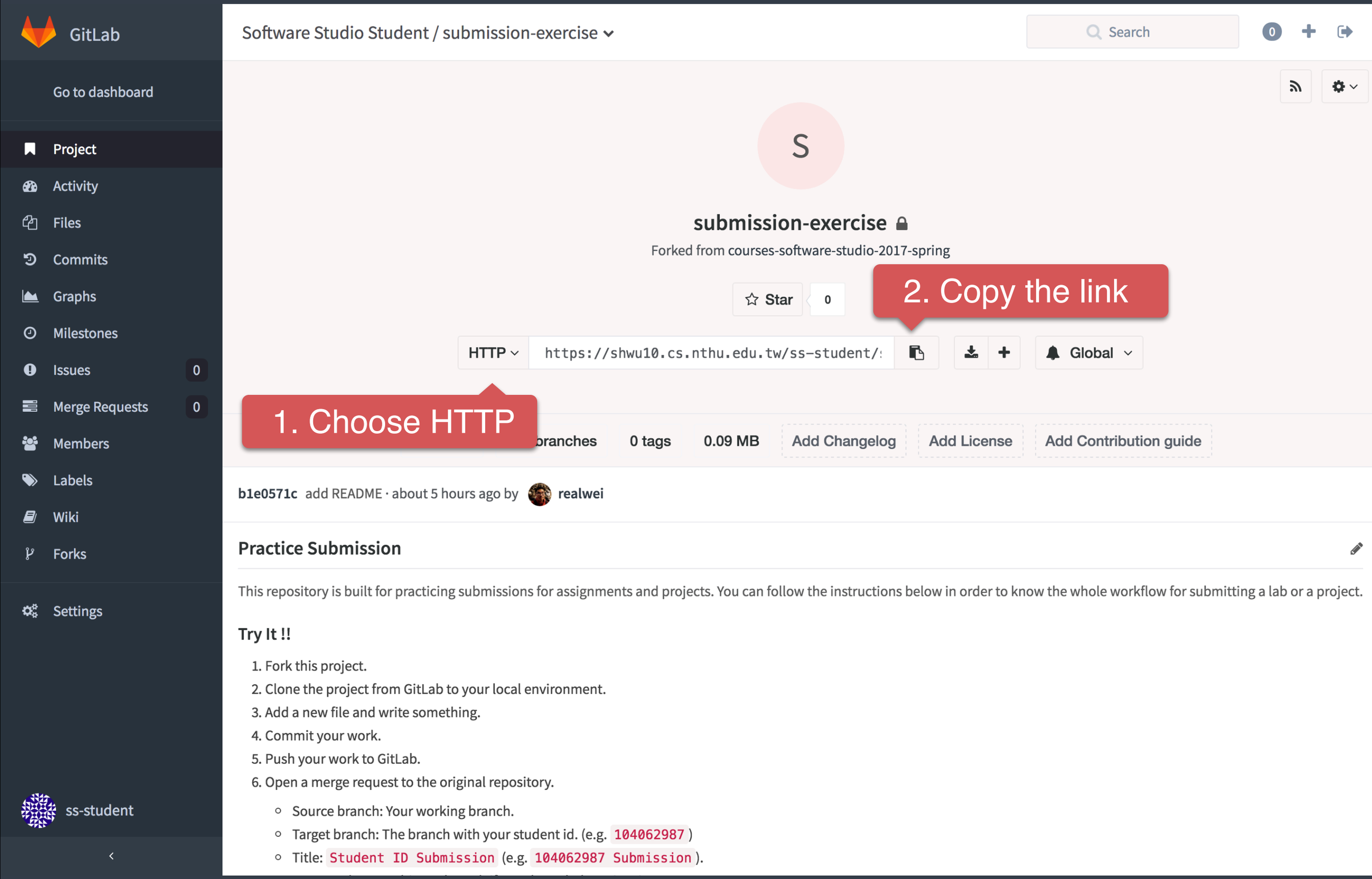
Save changes

ss-student

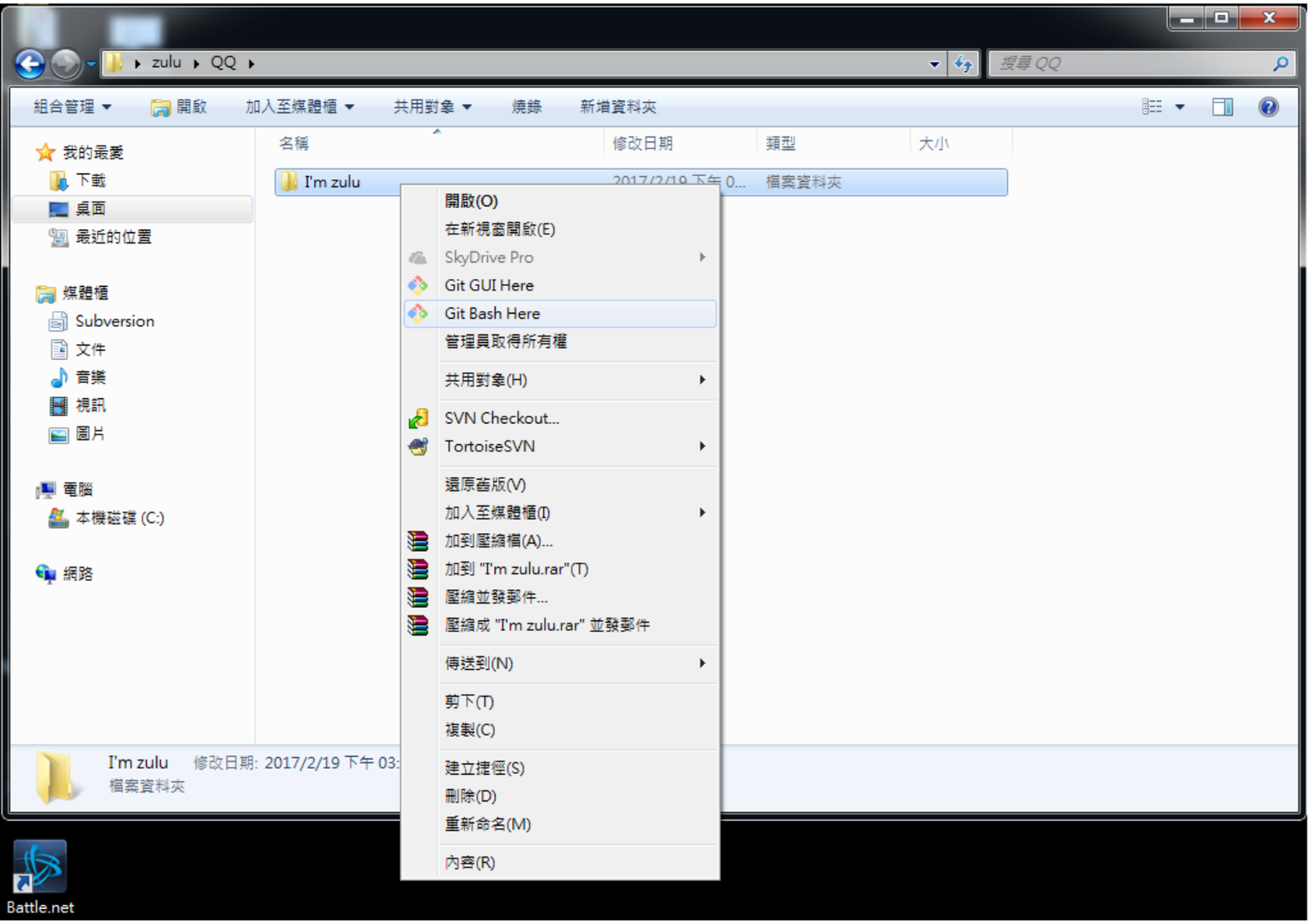
Workflow

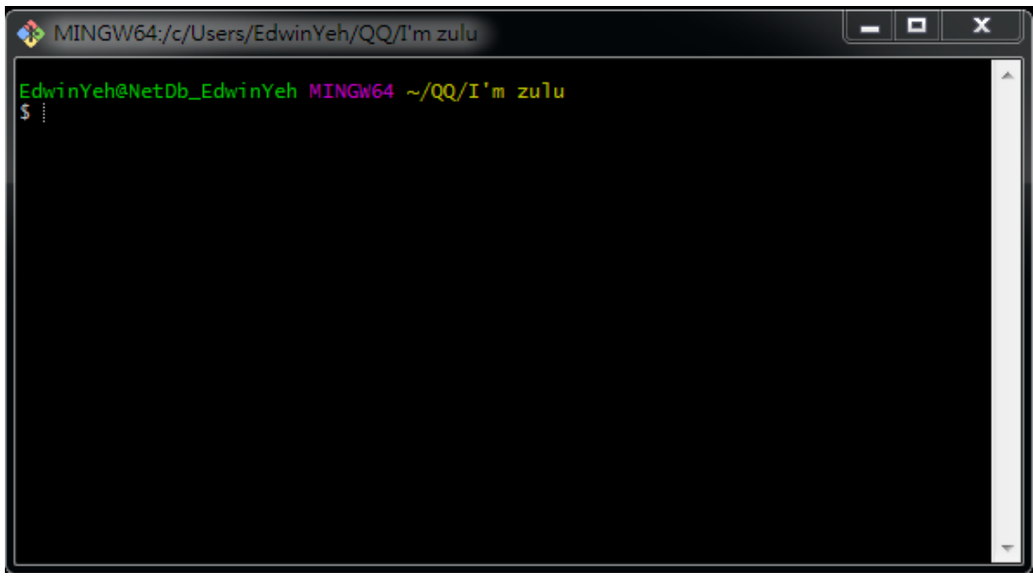
- For each lab, you should follow the workflow below

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
Software Studio Student / submission-exercise
Search
Go to dashboard
Project
Activity
Files
Commits
Graphs
Milestones
Issues 0
Merge Requests 0
Members
Labels
Wiki
Forks
Settings
ss-student
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
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).

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

~/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
```

1. -A means all files

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 0
- Merge Requests 0
- Members
- Labels
- Wiki
- Forks
- Settings

ss-student

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/s

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

Project
Activity
Files
Commits
Graphs
Milestones
Issues 0
Merge Requests 0
Members
Labels
Wiki
Forks

Settings

ss-student

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

GitLab

Go to dashboard

- Project
- Activity
- Files
- Commits
- Graphs
- Milestones
- Issues 0
- Merge Requests 0
- Members
- Labels
- Wiki
- Forks
- Settings

ss-student

Software Studio Student / submission-exercise · Merge Requests

Search

New Merge Reques

3. Choose the branch you pushed in your repo

4. Choose the branch named after your ID

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

GitLab

Go to dashboard

- Project
- Activity
- Files
- Commits
- Graphs
- Milestones
- Issues 0
- Merge Requests 0
- Members
- Labels
- Wiki
- Forks
- Settings

ss-student

Software Studio Student / submission-exercise · Merge Requests

Search 0

New Merge Request

From **ss-student/submissio** **sion-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

7. If everything is OK, submit your merge request

Cancel

Commits 1 | Changes 1

Most recent commits displayed first

18 Feb, 2017

1 commit

Finish lab1

Software Studio Student authored 24 minutes ago

14e22e1c

Browse Files »

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)

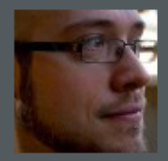
- fancier-refbar-changes
- fancy-responsive-refbar-it... 42↗ 99+
- graph-color-test
- hopscotch 24↗ 99+
- init-repo-gitignore-typeahead
- invite-system 6↗ 99+↓
- jars-view-file-history
- master 5↙
- remote-panel-redesign 15↗ 13↓
- settings-theme-styling
- view-file-history 24↗ 99+

REMOTE (6/41)

- Jeff-Schinella (0/1)
- Jordan-Wallet (0/7)
- Justin-GK (0/1)
- Ken-Price (0/2)
- Kyle-Smith (2/8)
- Max-Korp (0/2)
- Sjepan-Rajko (0/8)
- ayresa (0/3)
- cbargren (0/5)
- origin (4/4)

TAGS (99/99)

	Fix un/stageall and stashing
	Keep rename detection stage/unstage all
0.1.40	Bump to version 0.1.40
	Merge pull request #597 from johnhaley81/fix-dispatc...
	Merge pull request #594 from Mr-Wallet/nicer-ref-nam...
	Fix `waitFor` bug in dispatcher
	Merge pull request #596 from johndavidsparrow/gh-p...
	Revised custom variable script and switch
	Merge pull request #595 from johndavidsparrow/gh-p...
	Resolved edge case where RefNodes could overlap
	/universe removal of in-app invite wording
0.1.39	Bump to version 0.1.39
	Merge pull request #591 from Mr-Wallet/fix-graph-ref-...
	Merge pull request #590 from srajko/div-be-gone
	Merge pull request #588 from Mr-Wallet/friendlier-app...
	Merge pull request #568 from Mr-Wallet/nicer-ref-nam...
	Merge pull request #589 from johndavidsparrow/gh-p...
	Merge pull request #592 from implausible/FixNSFW
	JS tidy up in form-validation.js
	Javascript update for /universe
	/universe page
	added maxwait to updateworkdir debounce
	Update NSFW for memory leak
	Fix NSFW segfault
	Fix flickering GraphRefColumn every time ... 6 days ago
	Preventing page reload on default pull click
	Eliminate console spam when conflicts exist in a statel...
	Upgrading to react-bootstrap v0.24.5



Commit: cca151e6b9e32c3f9209c25131706740050
Parent: 8efe30a11761983173f844900fa5ec5c6be2
Author: John Haley <johnh@axosoft.com>
Author Date: September 30th 2015, 2:54 pm

Bump to version 0.1.40

0 added 0 deleted 2 modified

npm-shrinkwrap.json

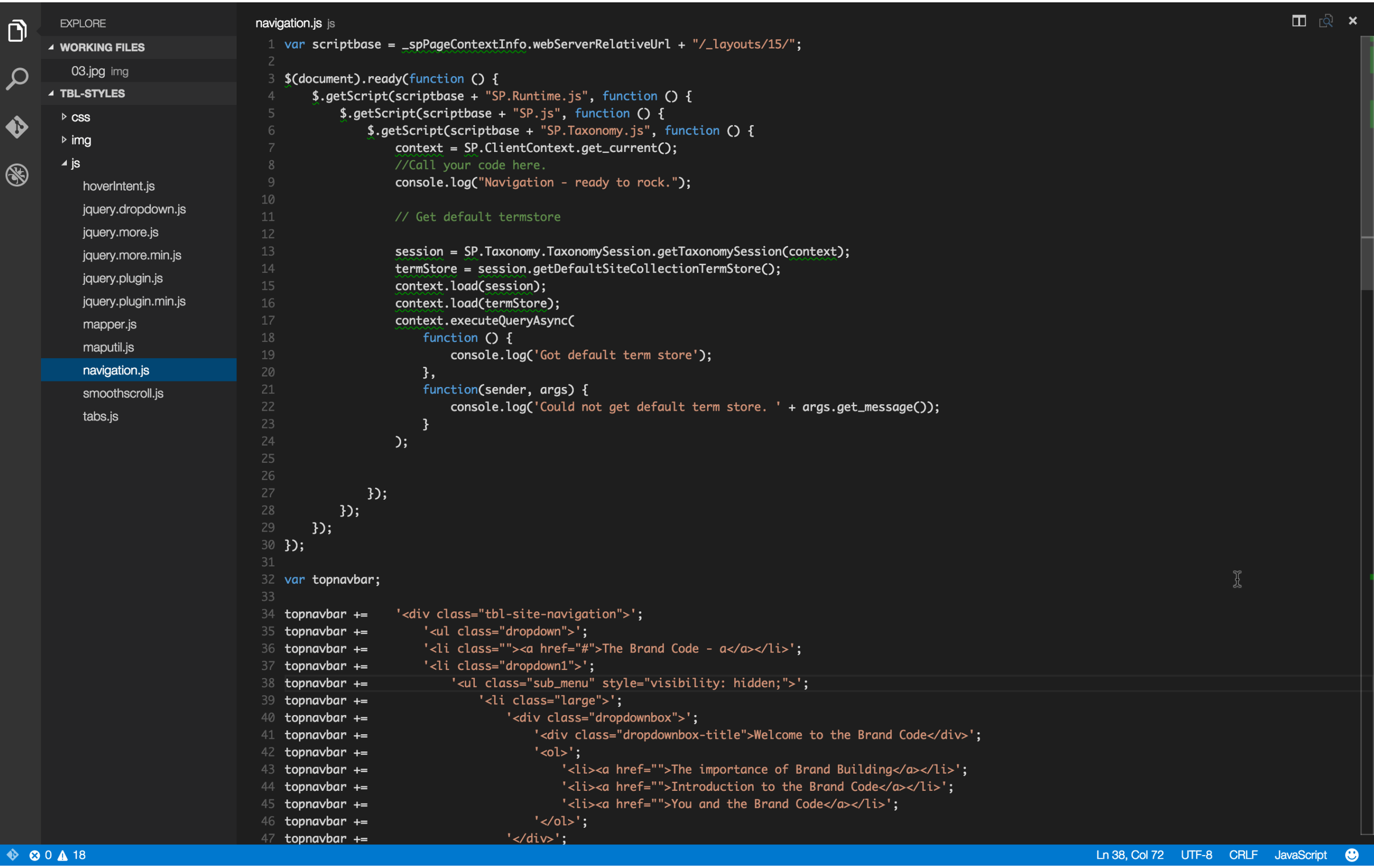
```
@@ -1,6 +1,6 @@
 1  1  {
 2  2    "name": "gitkraken",
-3     "version": "0.1.39",
    +3  "version": "0.1.40",
 4  4    "dependencies": {
 5  5      "atom-keymap": {
 6  6        "version": "5.1.11",
```

package.json

```
@@ -1,7 +1,7 @@
 1  1  {
 2  2    "name": "gitkraken",
 3  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

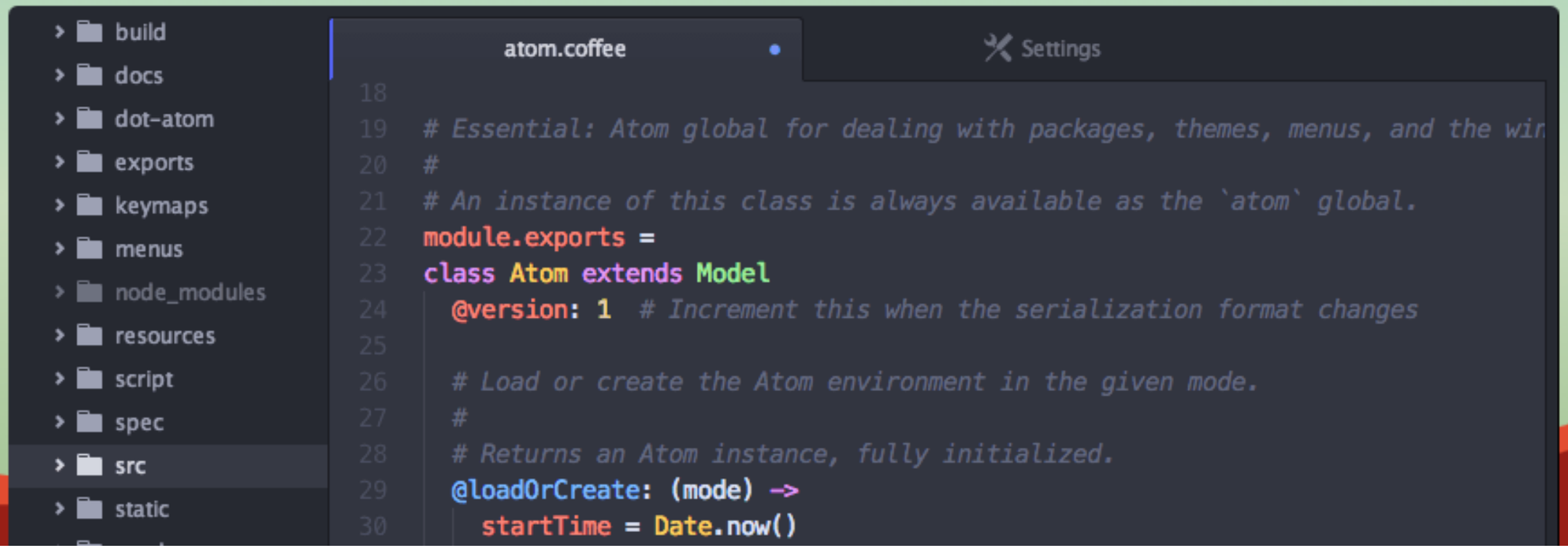

VS Code


EXPLORE
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 js
var scriptbase = _spPageContextInfo.webServerRelativeUrl + "/_layouts/15/";
$(document).ready(function () {
$.getScript(scriptbase + "SP.Runtime.js", function () {
$.getScript(scriptbase + "SP.js", function () {
$.getScript(scriptbase + "SP.Taxonomy.js", function () {
context = SP.ClientContext.get_current();
//Call your code here.
console.log("Navigation - ready to rock.");
// Get default termstore
session = SP.Taxonomy.TaxonomySession.getTaxonomySession(context);
termStore = session.getDefaultSiteCollectionTermStore();
context.load(session);
context.load(termStore);
context.executeQueryAsync(
function () {
console.log('Got default term store');
},
function(sender, args) {
console.log('Could not get default term store. ' + args.get_message());
}
);
});
});
});
});
var topnavbar;
topnavbar += ' ';
topnavbar += '<ul class="dropdown">';
topnavbar += '<li class="">The Brand Code - a
';
topnavbar += '<li class="dropdown1">';
topnavbar += '<ul class="sub_menu" style="visibility: hidden;">';
topnavbar += '<li class="large">';
topnavbar += ' ';
topnavbar += ' Welcome to the Brand Code ';
topnavbar += '<ol>';
topnavbar += '
The importance of Brand Building
';
topnavbar += '
Introduction to the Brand Code
';
topnavbar += '
You and the Brand Code
';
topnavbar += '
';
topnavbar += ' ';
0 18
Ln 38, Col 72
UTF-8
CRLF
JavaScript

ATOM

A hackable text editor

for the 21st Century



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 mission

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