Lab 00
How to Survive & Introduction to Git

Software Studio
DataLab, CS, NTHU

Teaching Assistants

Shao-Che Feng

馮劭哲

Chun-Ping Fang

方竣平

Tzu-Ling Hsieh

謝紫翎

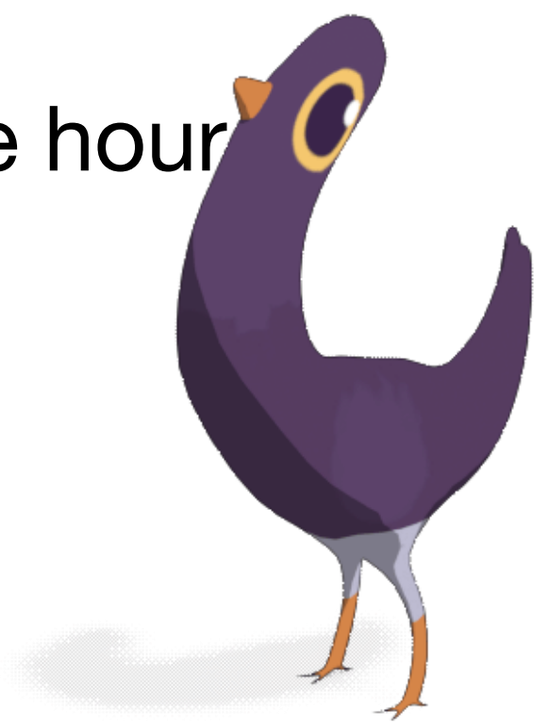
How to Find Us?

- Office Hour (TAs)
  - Tue. 1:30pm-3:30pm at Delta 729
  - Thur. 10:00am-12:00pm at Delta 729
- Email
  - ssta2025@datalab.cs.nthu.edu.tw
- Online Forum
  - eeclass

If I have Question?

If I have Question?

- Always Google or ask Chatgpt first!
  - Learn how to google and generative AI is important.
- If you try your best but still can’t catch it.
  - Feel free to ask us on eeclass or office hour



Today’s exercise

- Install Git command line tool in your computer
  - Follow appendix “Git Command-line Tool Installation”.
- Try to submit in GitLab.
- Set up flutter environment (If we have time)

Outline

- General Rule
- Introduction to Git
- Version control
- Branch and merge
- How to Submit Your Code to Gitlab
- Tools & References

Outline

- General Rule
- Introduction to Git
- Version control
- Branch and merge
- How to Submit Your Code to Gitlab
- Tools & References

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

Outline

- General Rule
- Introduction to Git
- Version control
- Branch and merge
- How to Submit Your Code to Gitlab
- Tools & References

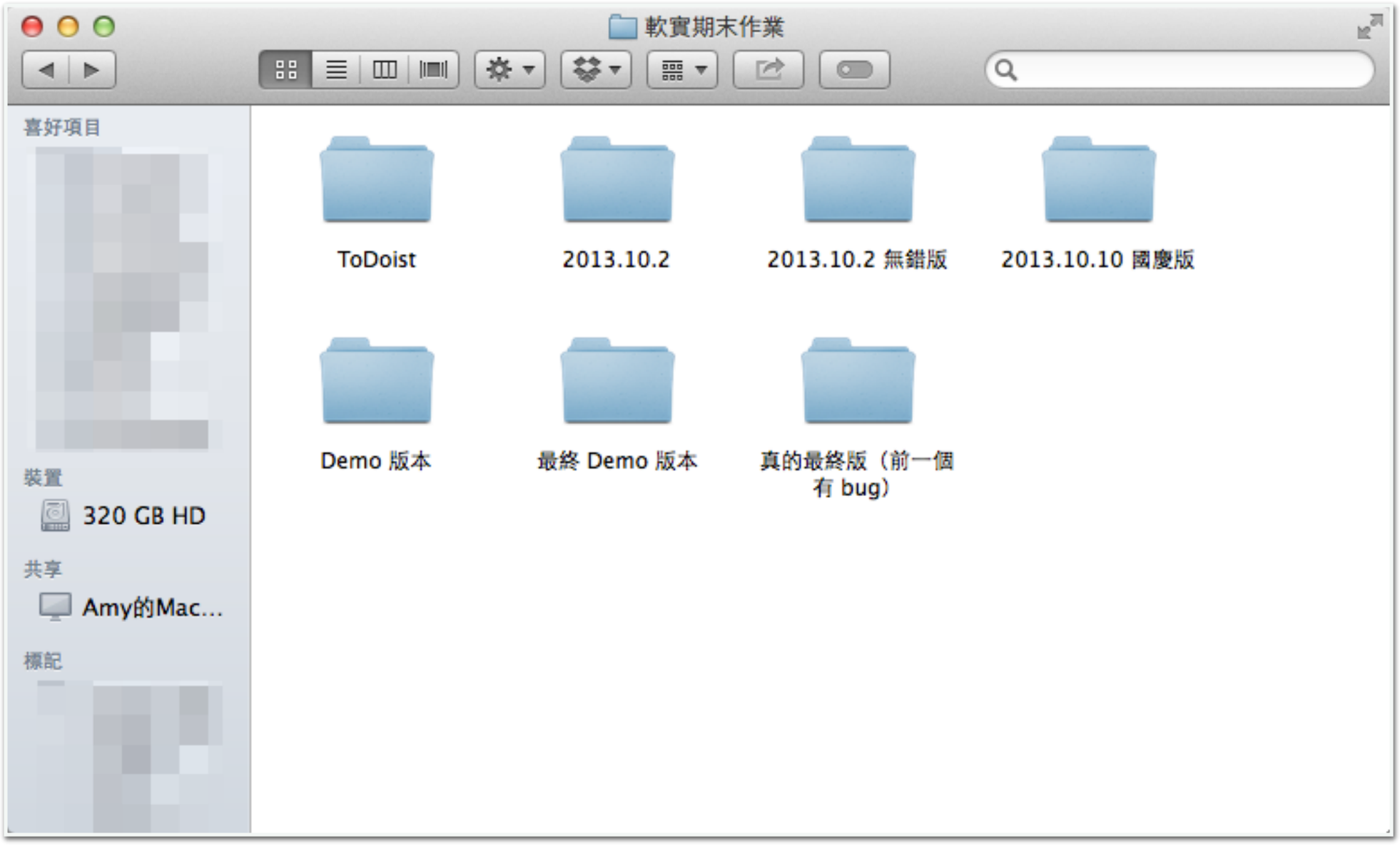
Why should we use version control?

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

Why use version control system?

- 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.

Students' VCS

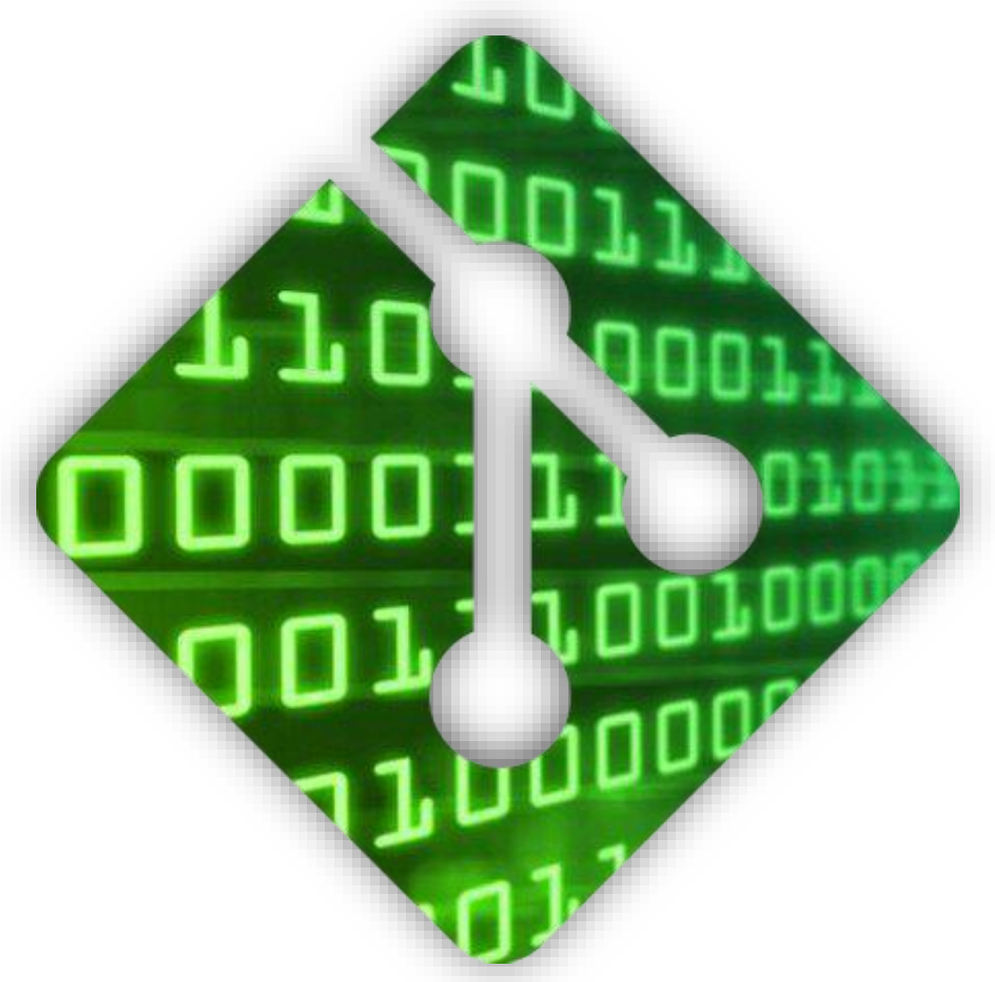


Git

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.

Be Professional



Outline

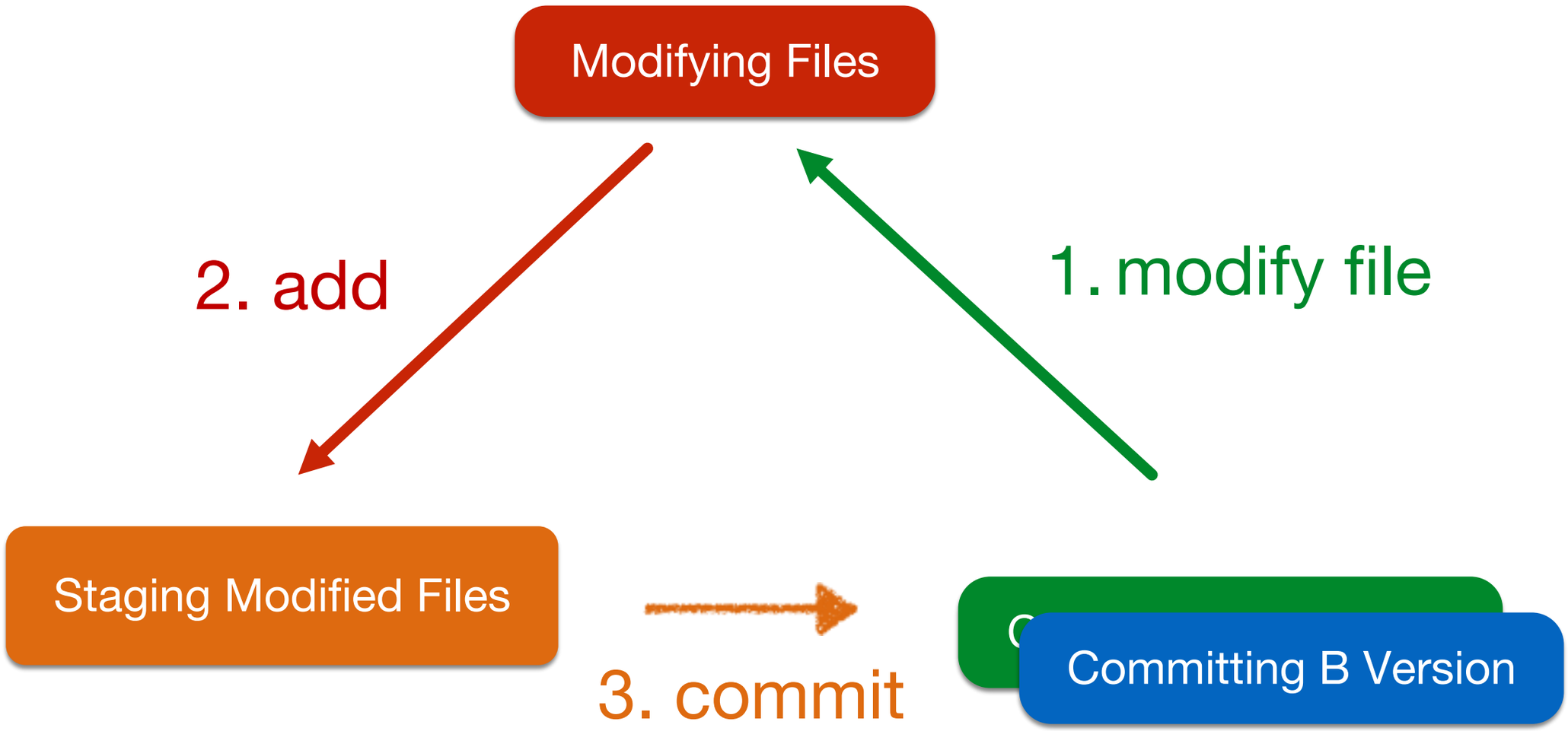
- General Rule
- Introduction to Git
- Version control
- Branch and merge
- How to Submit Your Code to Gitlab
- Tools & References

Git VCS (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 VCS (2/2)

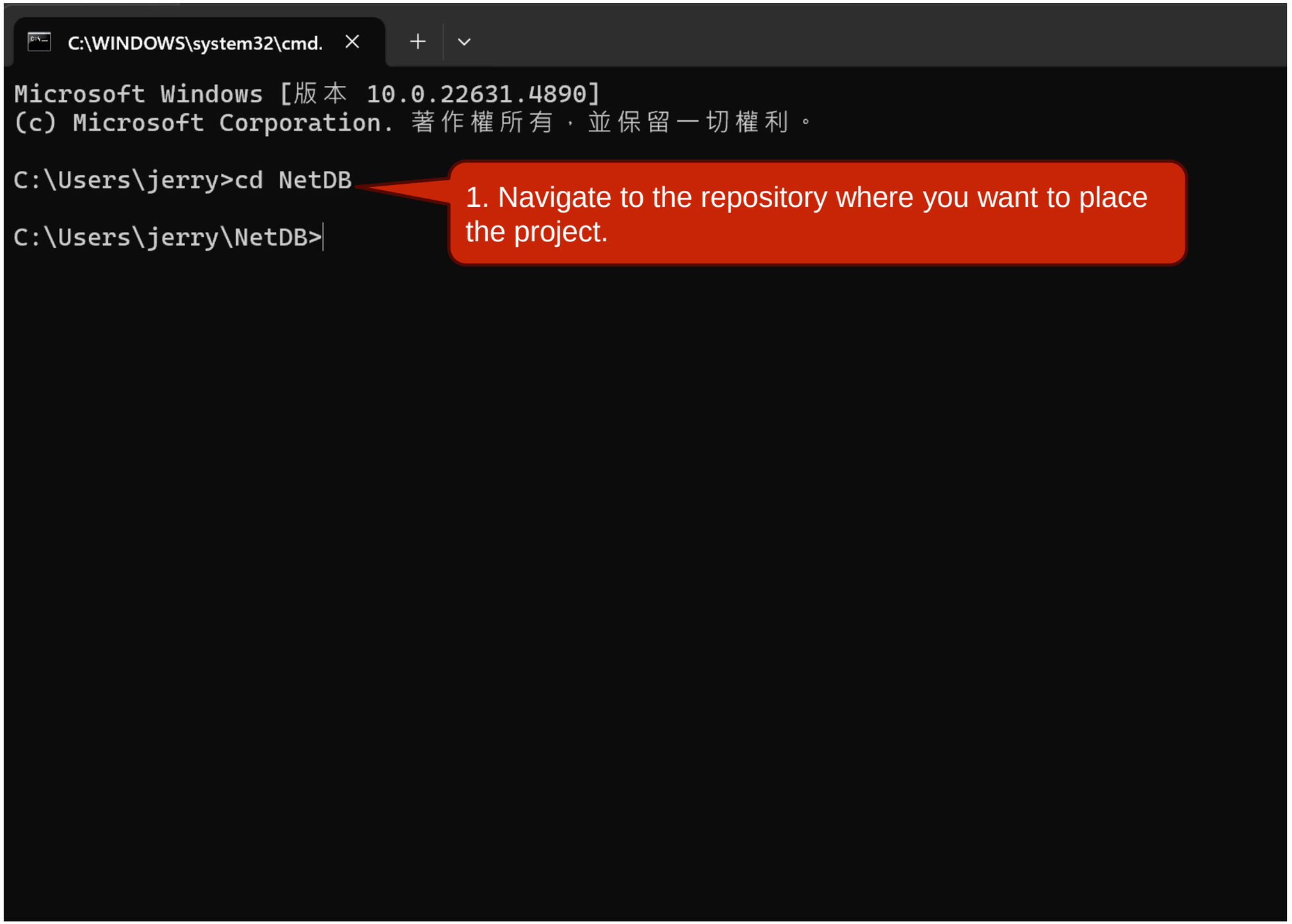

Modifying Files
2. add
1. modify file
Staging Modified Files
3. commit
Committing B Version

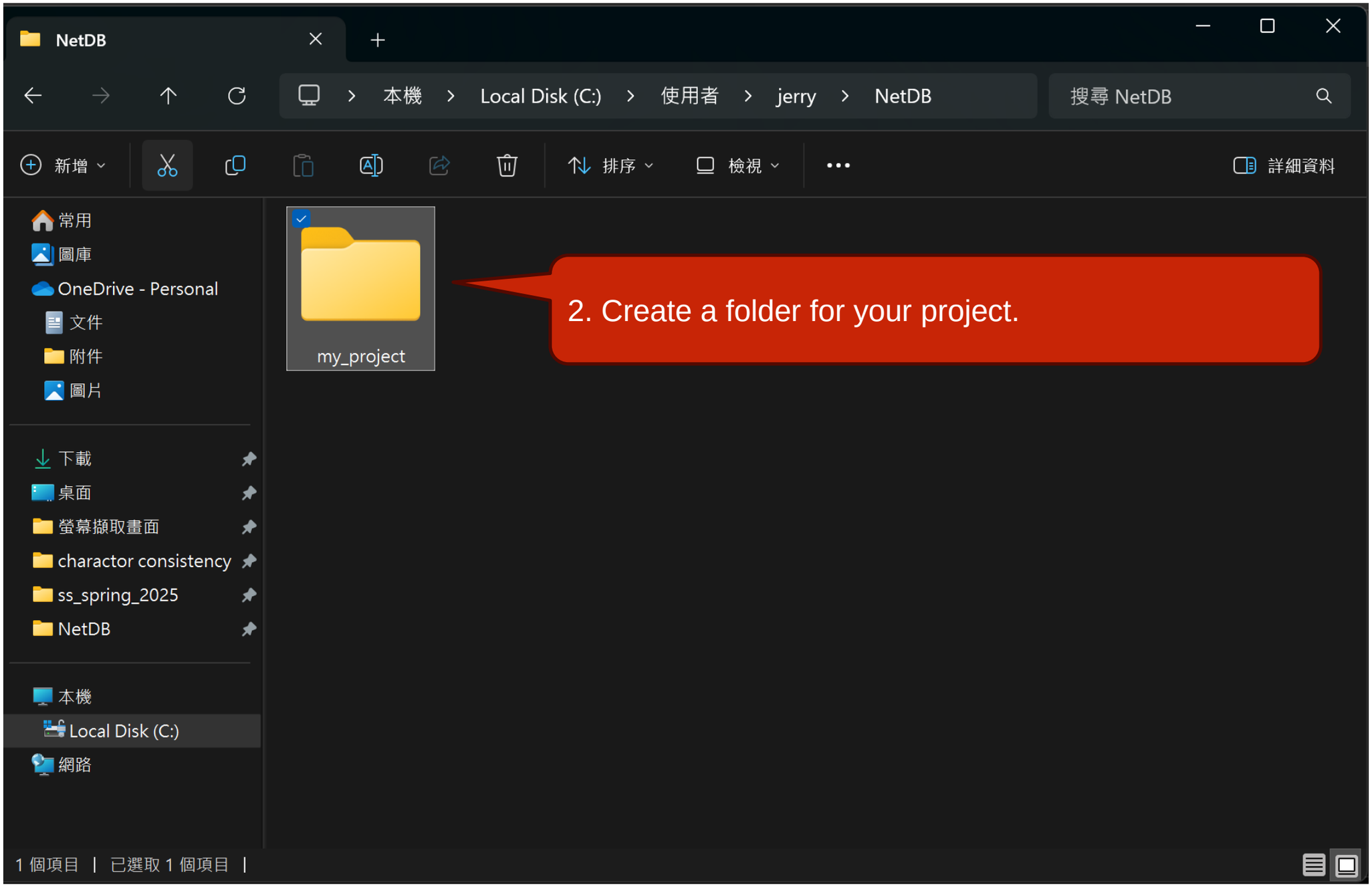
Basic Git Commands

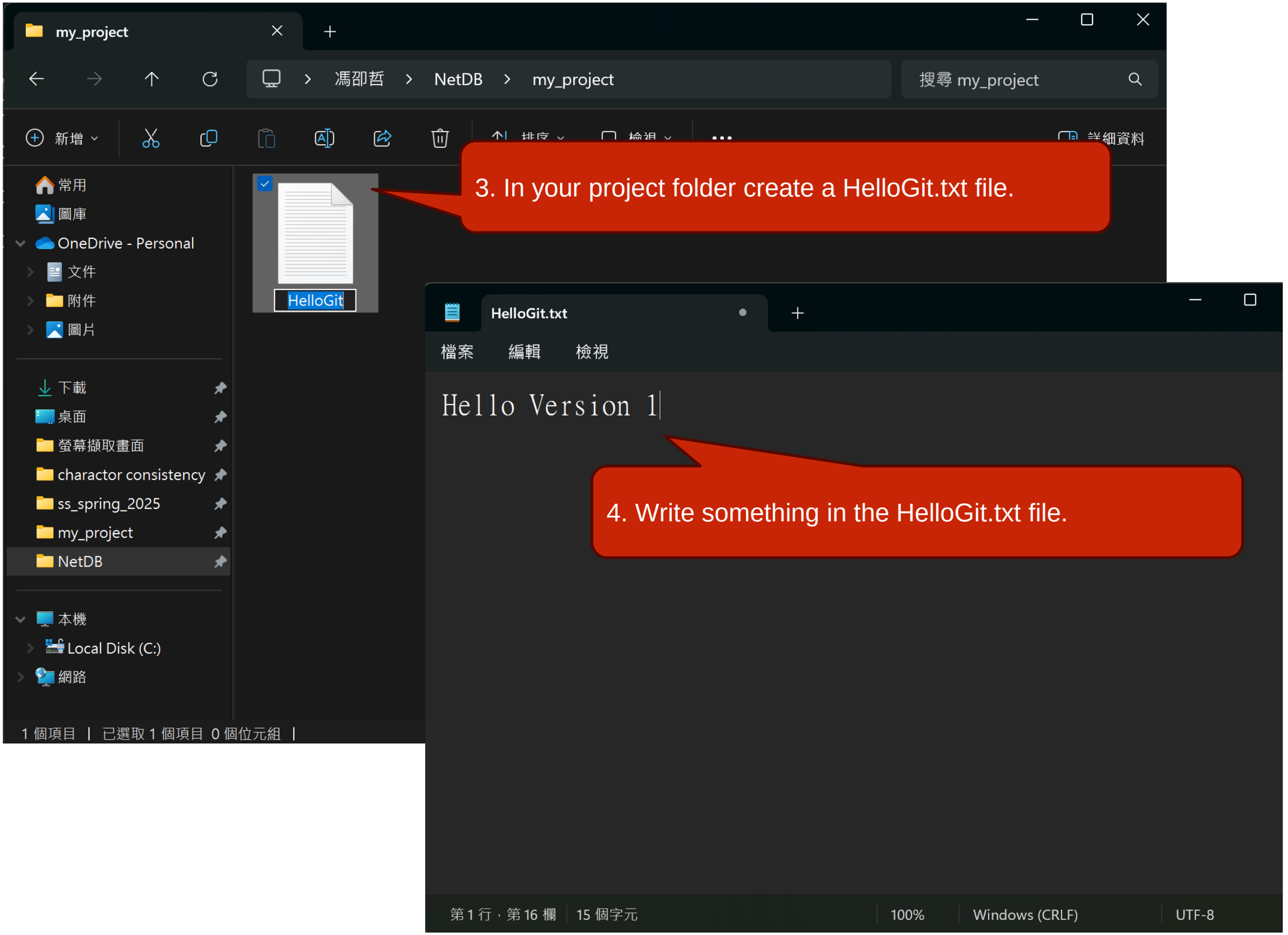
- **git init**
  - Initialize a repository at current directory.
- **git add [file_name] (* git add . * → means add all files)**
  - Add files to git repository and let git track them.
- **git commit -m "commit messages"**
  - Save the changes to the git repository and create snapshots of the files.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

scenario

- Create a project repository and initialize Git, and track the *HelloGit.txt* file for version control.


C:\WINDOWS\system32\cmd.
Microsoft Windows [版本 10.0.22631.4890]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\jerry>cd NetDB
C:\Users\jerry\NetDB>
1. Navigate to the repository where you want to place the project.


NetDB
本機
Local Disk (C:)
使用者
jerry
NetDB
搜尋 NetDB
新增
排序
檢視
詳細資料
常用
圖庫
OneDrive - Personal
文件
附件
圖片
下載
桌面
螢幕擷取畫面
charactor consistency
ss_spring_2025
NetDB
本機
Local Disk (C:)
網路
my_project
2. Create a folder for your project.
1 個項目 | 已選取 1 個項目 |


my_project
馮卻哲 › NetDB › my_project
搜尋 my_project
新增
詳細資料
常用
圖庫
OneDrive - Personal
文件
附件
圖片
下載
桌面
螢幕擷取畫面
charactor consistency
ss_spring_2025
my_project
NetDB
本機
Local Disk (C:)
網路
HelloGit
1 個項目 | 已選取 1 個項目 0 個位元組 |
3. In your project folder create a HelloGit.txt file.
HelloGit.txt
檔案 編輯 檢視
Hello Version 1
4. Write something in the HelloGit.txt file.
第 1 行，第 16 欄 15 個字元 100% Windows (CRLF) UTF-8

C:\WINDOWS\system32\cmd.

```
Microsoft Windows [版本 10.0.22631.4890]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\jerry>cd NetDB

C:\Users\jerry\NetDB>cd my_project

C:\Users\jerry\NetDB\my_project>dir
 磁碟區 C 中的磁碟是 Local Disk
 磁碟區序號：  8697-6595

 C:\Users\jerry\NetDB\my_project 的目錄

2025/02/18  下午 07:48    <DIR>          .
2025/02/18  下午 07:47    <DIR>          ..
2025/02/18  下午 07:49                15 HelloGit.txt.txt
               1 個檔案                15 位元組
               2 個目錄   190,638,534,656 位元組可用

C:\Users\jerry\NetDB\my_project>git init
Initialized empty Git repository in C:/Users/je

C:\Users\jerry\NetDB\my_project>
```

5. Navigate to the project fold.

6. Use the ‘’dir’’ command to list the contents of a folder. (if windows)

7. Use the ‘’git init’’ command to initialize a Git repository.

C:\WINDOWS\system32\cmd.

```
Microsoft Windows [版本 10.0.22631.4890]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\jerry>cd NetDB

C:\Users\jerry\NetDB>cd my_project

C:\Users\jerry\NetDB\my_project>dir
 磁碟區 C 中的磁碟是 Local Disk
 磁碟區序號：  8697-6595

 C:\Users\jerry\NetDB\my_project 的目錄

2025/02/18  下午 07:55    <DIR>          .
2025/02/18  下午 07:47    <DIR>          ..
2025/02/18  下午 07:55                 0 HelloGit.txt
               1 個檔案                0 位元組
               2 個目錄  190,626,844,672 位元組可用

C:\Users\jerry\NetDB\my_project>git init
Initialized empty Git repository in C:/Users/jerry/NetDB/my_project/.git/

C:\Users\jerry\NetDB\my_project>git add HelloGit.txt

C:\Users\jerry\NetDB\my_project>git commit -m "version 1"
[master (root-commit) 3c0e49e] version 1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 HelloGit.txt

C:\Users\jerry\NetDB\my_project>
```

8. Use the ‘’git add’’ command to add HelloGit.txt to staging file.

9. Use the “git commit –m” command to commit the change.

HelloGit.txt

檔案 編輯 檢視

Hello version 2

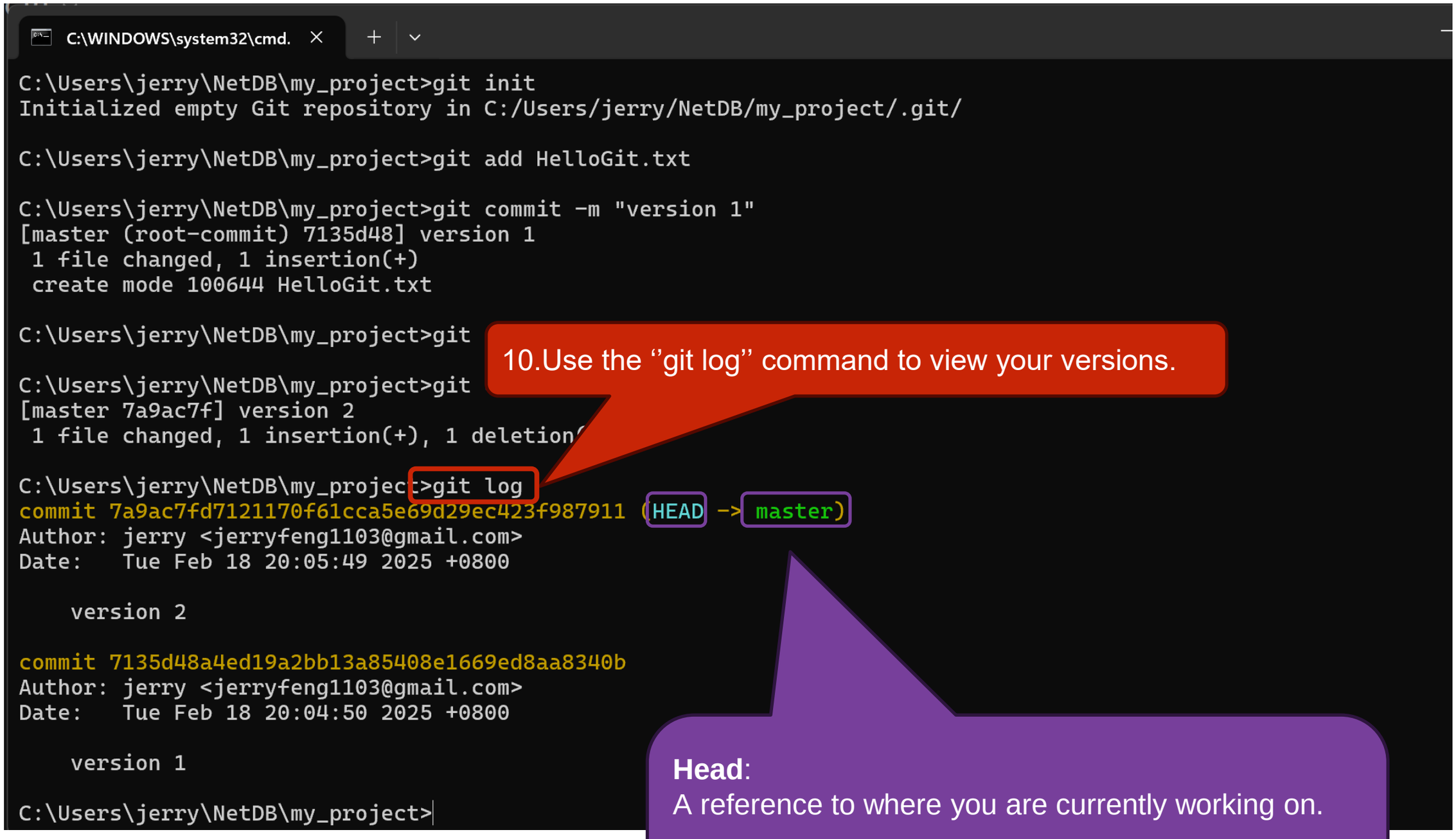
10. Modify the hello.txt and save the change.

C:\WINDOWS\system32\cmd.

```
C:\Users\jerry\NetDB\my_project>dir
 磁碟區 C 中的磁碟是 Local Disk
 磁碟區序號：  8697-6595

 C:\Users\jerry\NetDB\my_project 的目錄

2025/02/18  下午 08:03    <DIR>          .
2025/02/18  下午 08:03    <DIR>          ..
2025/02/18  下午 08:04                15 HelloGit.txt
               1 個檔案                15 位元組
               2 個目錄  190,627,364,864 位元組可用

C:\Users\jerry\NetDB\my_project>git init
Initialized empty Git repository in C:/Users/

C:\Users\jerry\NetDB\my_project>git add Hello

C:\Users\jerry\NetDB\my_project>git commit -m "versio
[master (root-commit) 7135d48] version 1
 1 file changed, 1 insertion(+)
 create mode 100644 HelloGit.txt

C:\Users\jerry\NetDB\my_project>git add HelloGit.txt

C:\Users\jerry\NetDB\my_project>git commit -m "version 2"
[master 7a9ac7f] version 2
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\jerry\NetDB\my_project>
```

11. Add It and commit again.

第 1 行，第 1 欄


C:\WINDOWS\system32\cmd.
C:\Users\jerry\NetDB\my_project>git init
Initialized empty Git repository in C:/Users/jerry/NetDB/my_project/.git/
C:\Users\jerry\NetDB\my_project>git add HelloGit.txt
C:\Users\jerry\NetDB\my_project>git commit -m "version 1"
[master (root-commit) 7135d48] version 1
1 file changed, 1 insertion(+)
create mode 100644 HelloGit.txt
C:\Users\jerry\NetDB\my_project>git
C:\Users\jerry\NetDB\my_project>git
[master 7a9ac7f] version 2
1 file changed, 1 insertion(+), 1 deletion
C:\Users\jerry\NetDB\my_project>git log
commit 7a9ac7fd7121170f61cca5e69d29ec423f987911 (HEAD -> master)
Author: jerry <jerryfeng1103@gmail.com>
Date: Tue Feb 18 20:05:49 2025 +0800
version 2
commit 7135d48a4ed19a2bb13a85408e1669ed8aa8340b
Author: jerry <jerryfeng1103@gmail.com>
Date: Tue Feb 18 20:04:50 2025 +0800
version 1
C:\Users\jerry\NetDB\my_project>
10.Use the ‘’git log’’ command to view your versions.
Head:
A reference to where you are currently working on.
Master:
The default branch in a Git repository, typically where the final, stable version of the project is maintained.

C:\WINDOWS\system32\cmd.

```
Switched to branch 'master'

C:\Users\jerry\NetDB\my_project>git log
commit 7a9ac7fd7121170f61cca5e69d29ec423f987911 (HEAD -> master)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 20:05:49 2025 +0800

    version 2

commit 7135d48a4ed19a2bb13a85408e1669ed8aa8340b
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 20:04:50 2025 +0800

    version 1

C:\Users\jerry\NetDB\my_project>git checkout 7135d48a4ed19a2bb13a85408e1669ed8aa8340b
Note: switching to '7135d48a4ed19a2bb13a85408e1669ed8aa8340b'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and yo
state without impacting any bra

If you want to create a new bra
do so (now or later) by using -

  git switch -c <new-branch-nam
```

Version ID

11. Go to specific version with the ‘’git checkout’’ command and version ID

HelloGit.txt

檔案 編輯 檢視

Hello Version 1

12. Reopen the HelloGit.txt find yourself back to version 1 !

C:\WINDOWS\system32\cmd.

```
commit 7a9ac7fd7121170f61cca5e69d29ec423f987911 (master)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 20:05:49 2025 +0800

    version 2

commit 7135d48a4ed19a2bb13a85408e1669ed8aa8340b (HEAD)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 20:04:50 2025 +0800

    version 1

C:\Users\jerry\NetDB\my_project>git checkout master
Previous HEAD position was 7135d48 version 1
Switched to branch 'master'

C:\Users\jerry\NetDB\my_project>git log
commit 7a9ac7fd7121170f61cca5e69d29ec423f987911 (HEAD -> master)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 20:05:49 2025 +0800

    version 2

commit 7135d48a4ed19a2bb13a85408e1669ed8aa8340b
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 20:04:50 2025 +0800

    version 1

C:\Users\jerry\NetDB\my_project>
```

13. Use the “git checkout master” command to go back to the final vesion of your project

HelloGit.txt

檔案 編輯 檢視

Hello version 2

14. Reopen the HelloGit.txt again. Back to the lastest version.

第 1 行，第 1 欄 | 15 個字元 | 100% | Windows (CRLF)

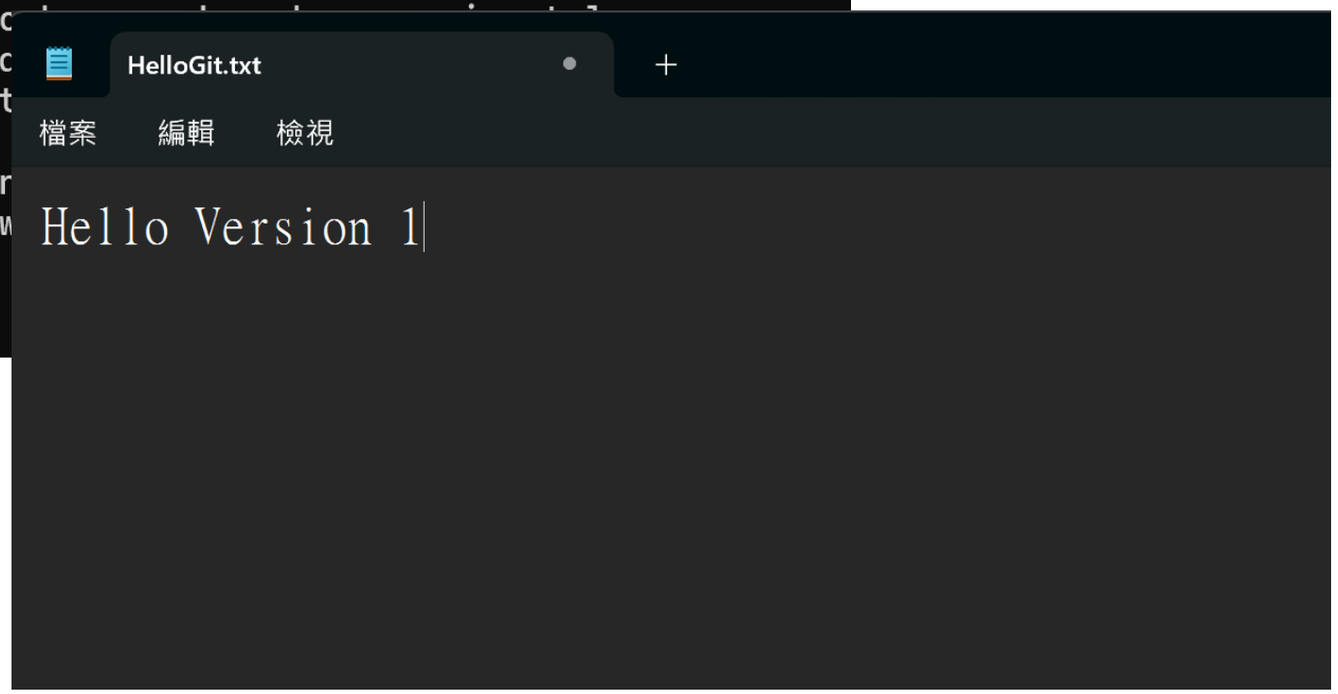
LIFE IS
TOO SHORT
TO TYPE THAT
VERSION ID*!*

which is 40 characters long…

```
C:\Users\jerry\NetDB\my_project>git log
commit 7a9ac7fd7121170f61cca5e69d29ec423f987911 (HEAD -> master)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 20:05:49 2025 +0800

    version 2

commit 7135d48a4ed19a2bb13a85408e1669ed8aa8340b
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 20:04:50 2025 +0800

    version 1

C:\Users\jerry\NetDB\my_project>git log --oneline
7a9ac7f (HEAD -> master) version 2
7135d48 version 1

C:\Users\jerry\NetDB\my_project>git checkout 7135d48
Note: switching to '7135d48'.

You are in 'detached HEAD' state. You can lo
changes and commit them, and you can discarc
state without impacting any branches by swit

If you want to create a new branch to retair
do so (now or later) by using -c with the sw

  git switch -c <new-branch-name>
```

15. Use the ‘’git log --oneline’’ command to get the shorter version ID.

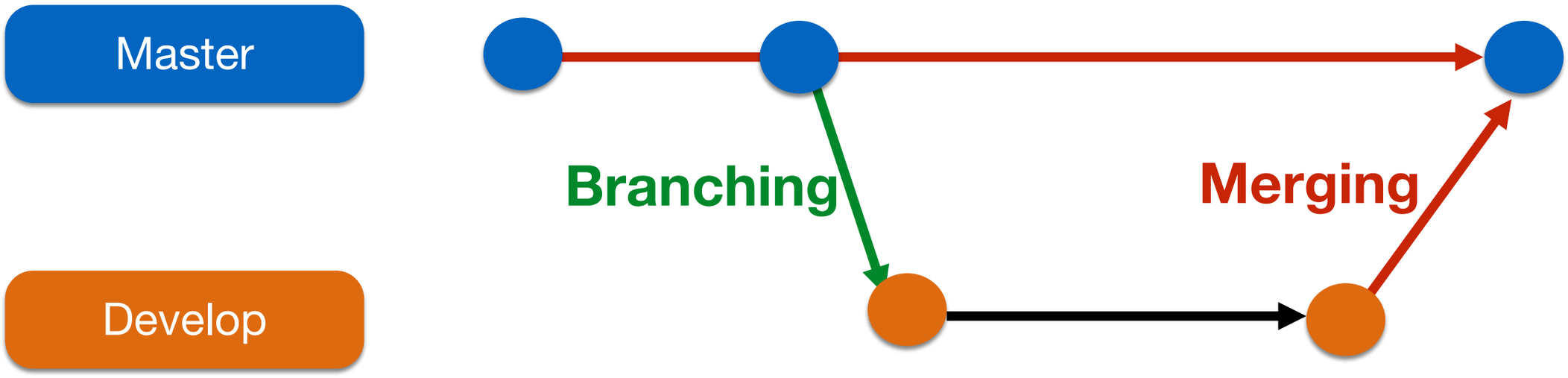
16. Use shorter version ID to check specific version.



Outline

- General Rule
- Introduction to Git
- Version control
- Branch and merge
- How to Submit Your Code to Gitlab
- Tools & References

Git Branch



Git commands (1/2)

- Branching steps
    - Creating a new branch

```
git branch [branch name]
```

    - Checking out the branch

```
git checkout [branch name]
```

Git commands (2/2)

- Merging steps
    - Checking out a branch to merge

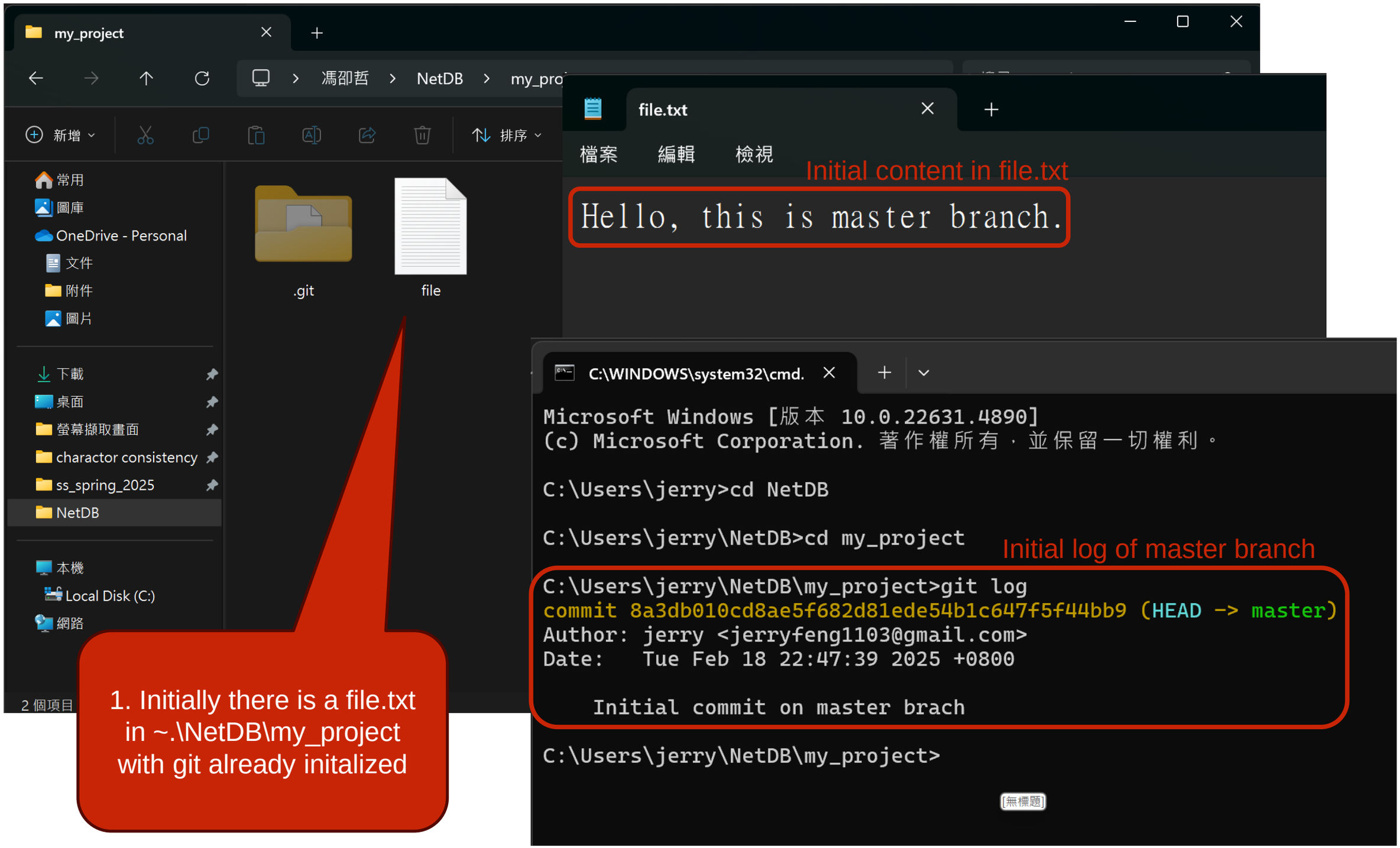
```
git checkout [branch 1 name]
```

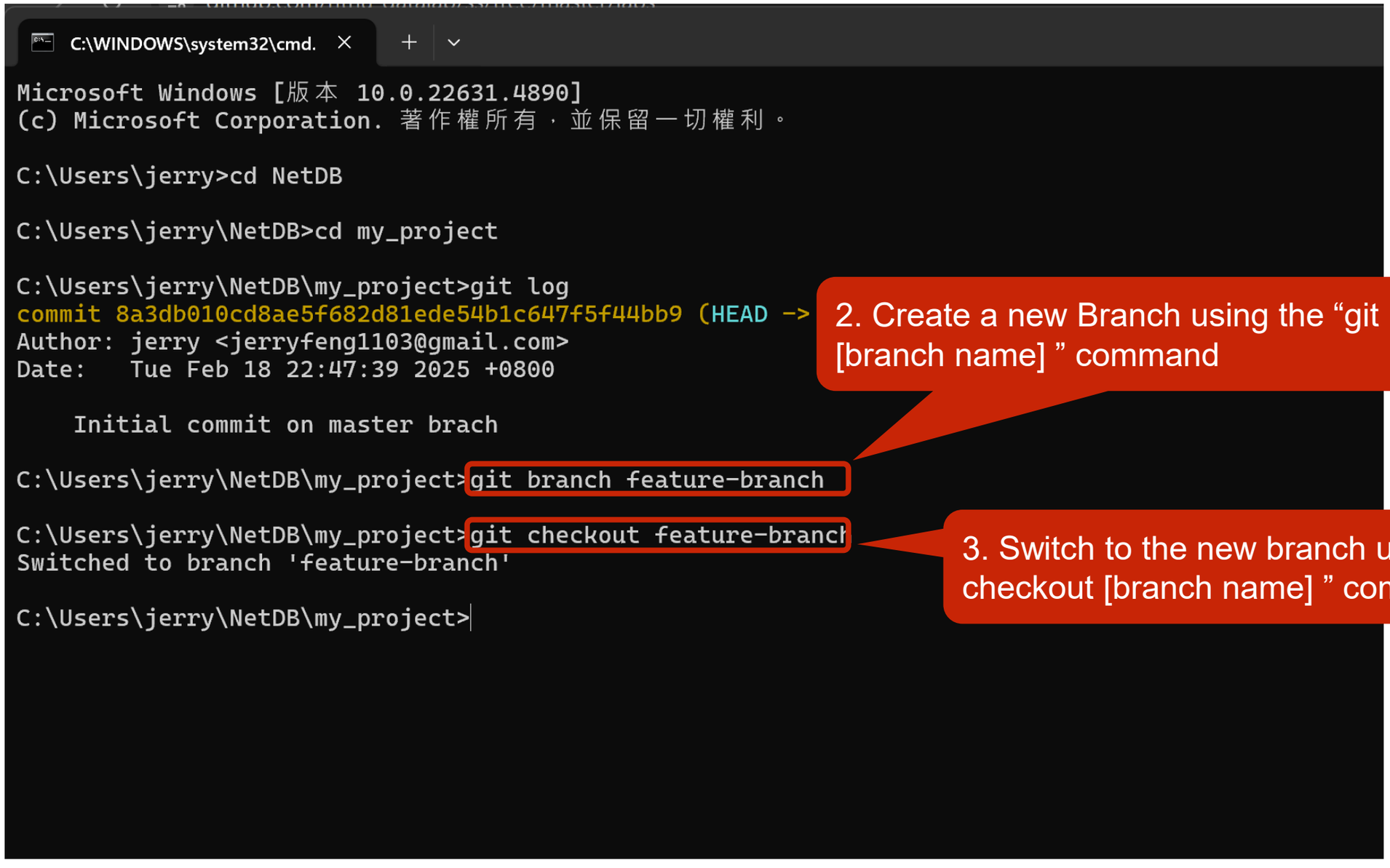
- Merging another branch

```
git merge [branch 2 name]
```

scenario

- Assume you are working on a new feature in a Git repository and want to create a separate branch work on it and then merge it back into master branch
-


my_project
馮卻哲
NetDB
新增
排序
常用
圖庫
OneDrive - Personal
文件
附件
圖片
下載
桌面
螢幕擷取畫面
charactor consistency
ss_spring_2025
NetDB
本機
Local Disk (C:)
網路
.git
file
2 個項目
1. Initially there is a file.txt in ~.\NetDB\my_project with git already initalized
file.txt
檔案
編輯
檢視
Initial content in file.txt
Hello, this is master branch.
C:\WINDOWS\system32\cmd.
Microsoft Windows [版本 10.0.22631.4890]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\jerry>cd NetDB
C:\Users\jerry\NetDB>cd my_project
Initial log of master branch
C:\Users\jerry\NetDB\my_project>git log
commit 8a3db010cd8ae5f682d81ede54b1c647f5f44bb9 (HEAD -> master)
Author: jerry <jerryfeng1103@gmail.com>
Date: Tue Feb 18 22:47:39 2025 +0800
Initial commit on master brach
C:\Users\jerry\NetDB\my_project>
[無標題]


C:\WINDOWS\system32\cmd.
Microsoft Windows [版本 10.0.22631.4890]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\jerry>cd NetDB
C:\Users\jerry\NetDB>cd my_project
C:\Users\jerry\NetDB\my_project>git log
commit 8a3db010cd8ae5f682d81ede54b1c647f5f44bb9 (HEAD ->
Author: jerry <jerryfeng1103@gmail.com>
Date: Tue Feb 18 22:47:39 2025 +0800
Initial commit on master brach
C:\Users\jerry\NetDB\my_project>git branch feature-branch
C:\Users\jerry\NetDB\my_project>git checkout feature-branch
Switched to branch 'feature-branch'
C:\Users\jerry\NetDB\my_project>
2. Create a new Branch using the “git branch [branch name] ” command
3. Switch to the new branch using the “git checkout [branch name] ” command

file.txt

檔案 編輯 檢視

```
Hello, this is the master branch.
This is a new feature.
```

4. Make Changes in the feature branch

C:\WINDOWS\system32\cmd.

```
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 23:01:19 2025 +0800

    Initial commit on master branch

C:\Users\jerry\NetDB\my_project>git branch feature-branch

C:\Users\jerry\NetDB\my_project>git checkout feature-branch
Switched to branch 'feature-branch'

C:\Users\jerry\NetDB\my_project>git add *

C:\Users\jerry\NetDB\my_project>git commit -m "Added a new feature"
[feature-branch dfd304e] Added a new feature
 1 file changed, 1 insertion(+)

C:\Users\jerry\NetDB\my_project>git log
commit dfd304ed1b565bf1c21f3c0a4e9f7cfd0b7a8ff8 (HEAD -> feature-branch)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 23:03:46 2025 +0800

    Added a new feature

commit fb903f40d1c918951ca48ce41347c805ab203a21 (master)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 23:01:19 2025 +0800

    Initial commit on master branch

C:\Users\jerry\NetDB\my_project>
```

5. Add and commit the check in feature branch. (use ‘’git add *’’ to stage all files in the current directory)

Since we already switch to feature branch. ‘’git log” will show the log in feature branch

file.txt

檔案 編輯 檢視

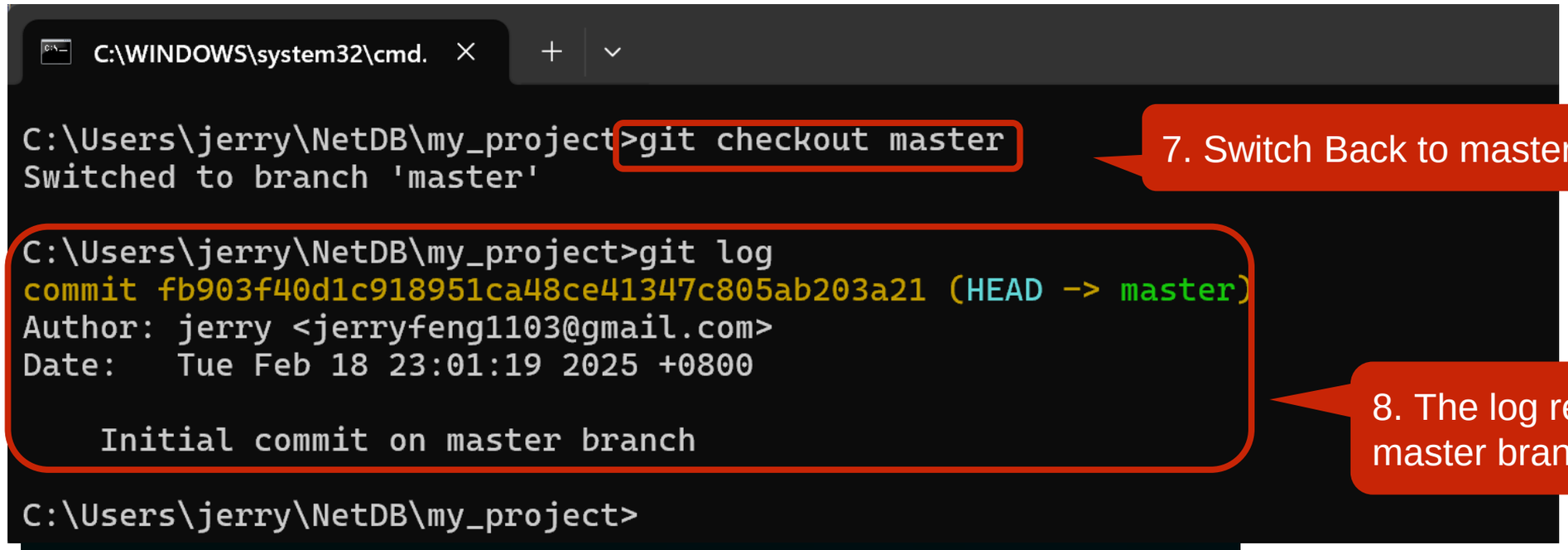
```
Hello, this is the master branch.
This is a new feature.
This is a second feature.
```

6. Perform another change, repeat the previous steps, and observe the Git log.

C:\WINDOWS\system32\cmd.

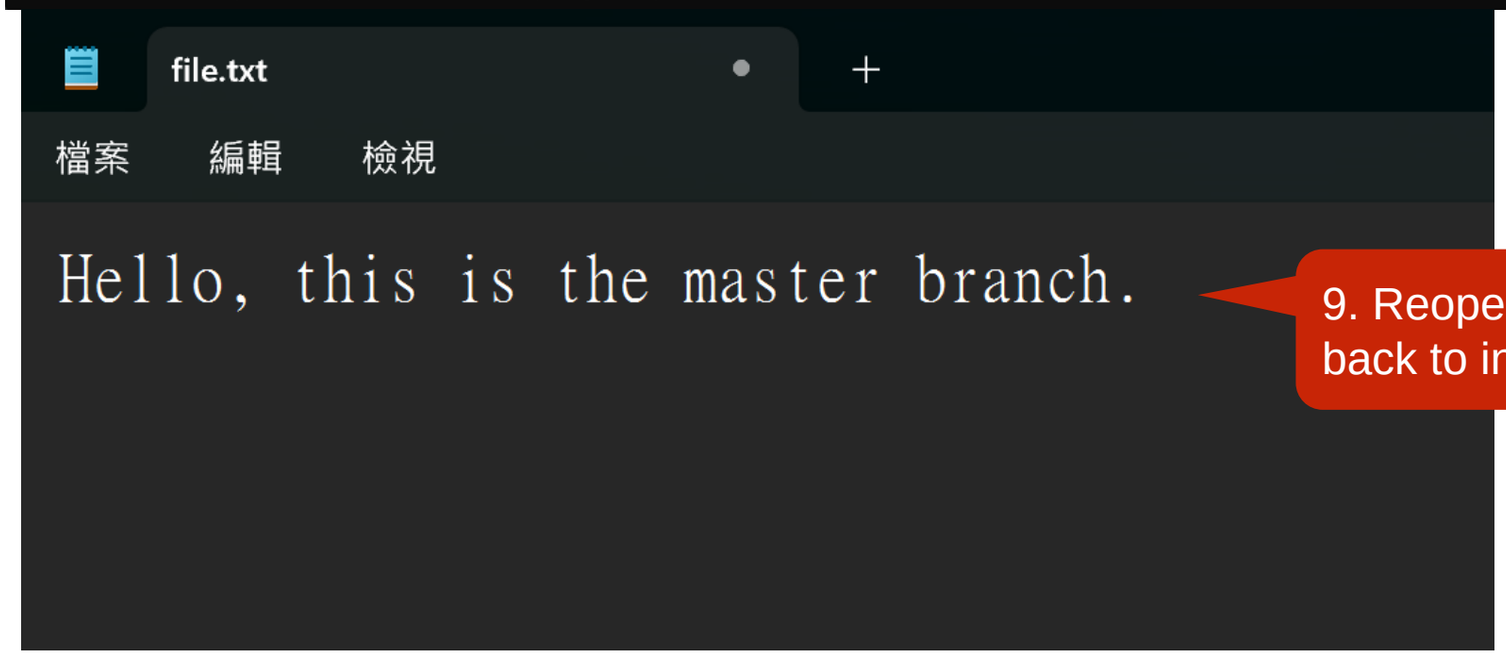
```
Date:   Tue Feb 18 23:01:19 2025 +0800

    Initial commit on master branch

C:\Users\jerry\NetDB\my_project>git add *

C:\Users\jerry\NetDB\my_project>git commit -m "Addend a second feature"
[feature-branch 22d4ba2] Addend a second feature
 1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\jerry\NetDB\my_project>git log
commit 22d4ba229447145c0565807a898433f7bf8cbb6e (HEAD -> feature-branch)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 23:09:08 2025 +0800

    Addend a second feature

commit dfd304ed1b565bf1c21f3c0a4e9f7cfd0b7a8ff8
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 23:03:46 2025 +0800

    Added a new feature

commit fb903f40d1c918951ca48ce41347c805ab203a21 (master)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 23:01:19 2025 +0800

    Initial commit on master branch

C:\Users\jerry\NetDB\my_project>
```

Reopen file.txt, and the content reverts to its initial state.



7. Switch Back to master branch

8. The log remains unchanged in the master branch."



9. Reopen the file.txt and the content go back to initial

file.txt

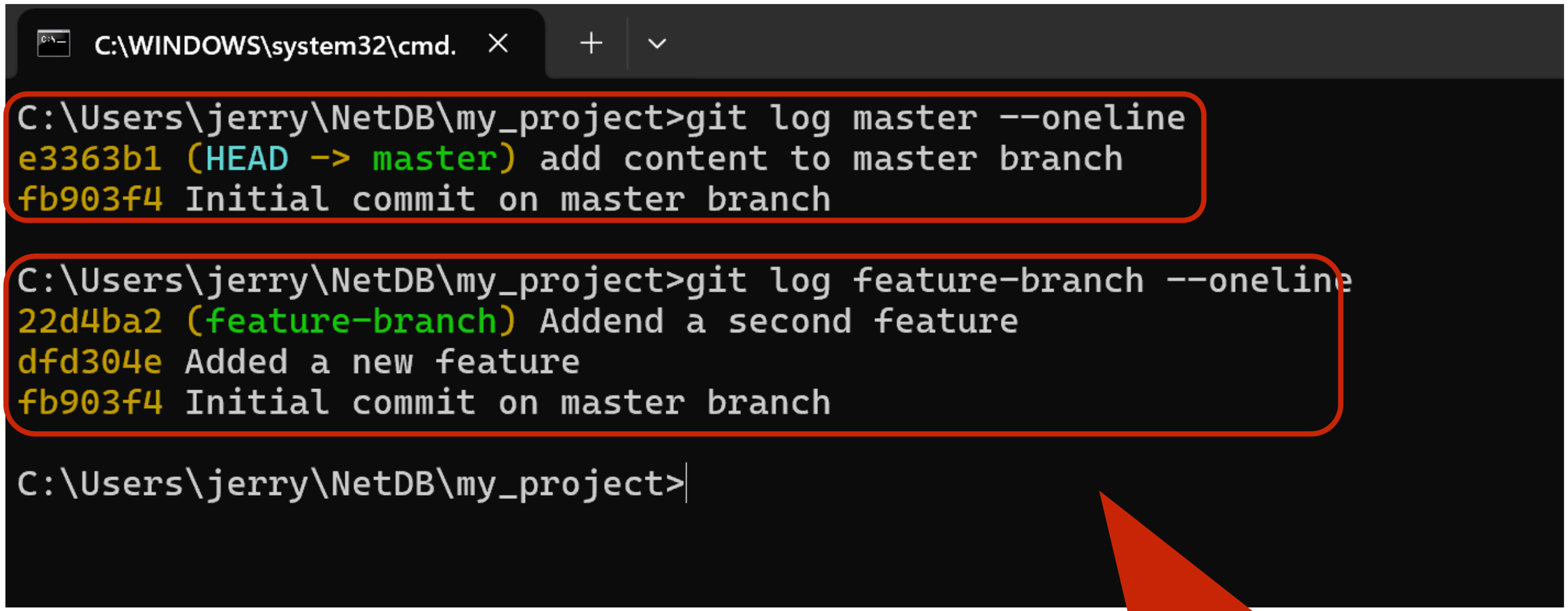
檔案 編輯 檢視

Hello, this is the master branch.
How are you?

10. Try to also make change in master branch and commit the change

C:\WINDOWS\system32\cmd.

```
C:\Users\jerry\NetDB\my_project>git log
commit fb903f40d1c918951ca48ce41347c805ab203a21 (HEAD -> master)
Author: jerry <jerryfeng1103@gmail.com>
Date:   Tue Feb 18 23:01:19 2025 +0800

    Initial commit on master branch

C:\Users\jerry\NetDB\my_project>git add *

C:\Users\jerry\NetDB\my_project>git commit -m "add content to master branch"
[master e3363b1] add content to master branch
 1 file changed, 1 insertion(+)
```



As observed in the log, a commit in one branch will not affect another branch.

C:\WINDOWS\system32\cmd.

```
C:\Users\jerry\NetDB\my_project>git checkout master
Already on 'master'

C:\Users\jerry\NetDB\my_project>git merge feature-branch
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\jerry\NetDB\my_project>
```

11. Ensure to move back to master branch (or any target branch you want to merge with)

12. Use the “git merge [branch name]” to merge the branch

Note that there is a merge conflict

file.txt

檔案 編輯 檢視

```
Hello, this is the master branch.
<<<<<<< HEAD
How are you?
=======
This is a new feature.
This is a second feature.
>>>>>>> feature-branch
```

13. Reopen the file.txt file then you found that there is merge conflict mark

<<<<<<< HEAD:
Show the change from your target branch.

======:
Separator between the conflicting change

>>>>>>> feature-branch
The changes from the branch you are trying to marge with the target branch

file.txt

檔案 編輯 檢視

```
Hello, this is the master branch
How are you?
This is a new feature.
This is a second feature.
```

14. Decide which version to keep and edit the file to resolve a marge conflict and save the change. In this case I reserve all the version.

C:\WINDOWS\system32\cmd.

```
Already on 'master'

C:\Users\jerry\NetDB\my_project>git merge feature-branch
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit t

C:\Users\jerry\NetDB\my_project>git add *

C:\Users\jerry\NetDB\my_project>git commit -m "Resolved merge conflict in file.txt"
[master 2aed399] Resolved merge conflict in file.txt

C:\Users\jerry\NetDB\my_project>
```

15. Add and commit again

Outline

- General Rule
- Introduction to Git
- Version control
- Branch and merge
- 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)

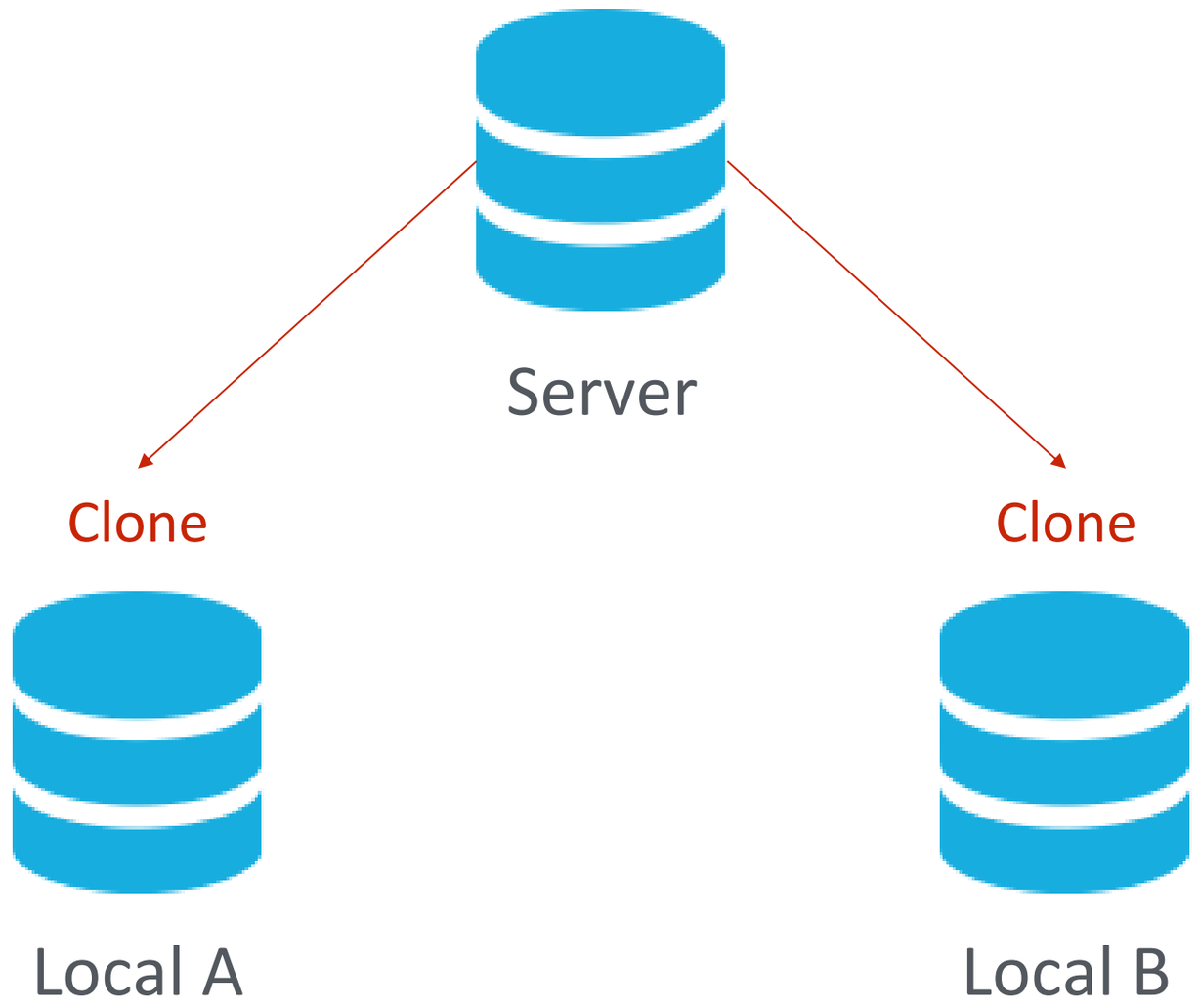
1. If you don’t have the project, *clone* (download) the repository from the server.

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)

Server

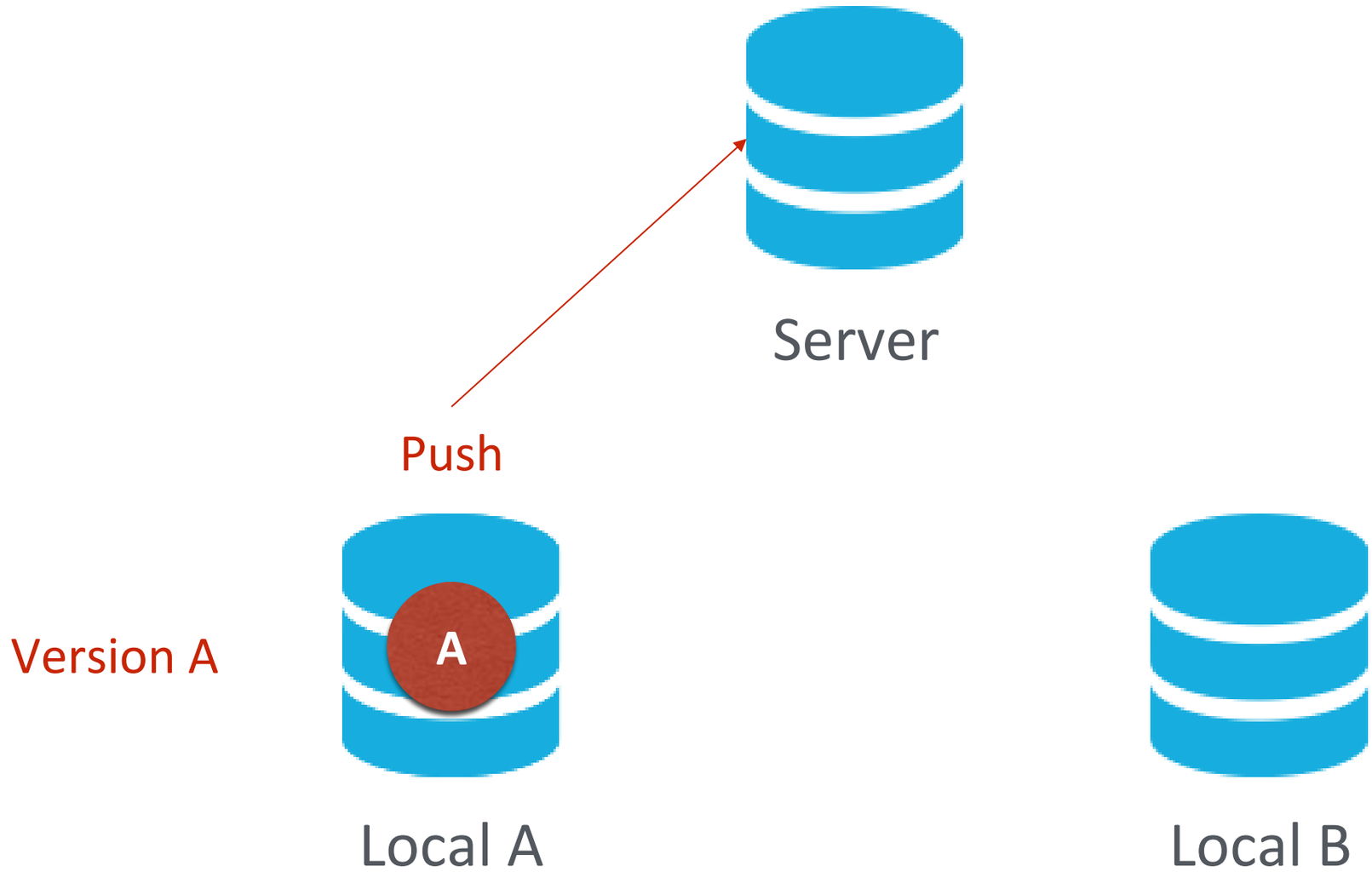
Commit

Version A

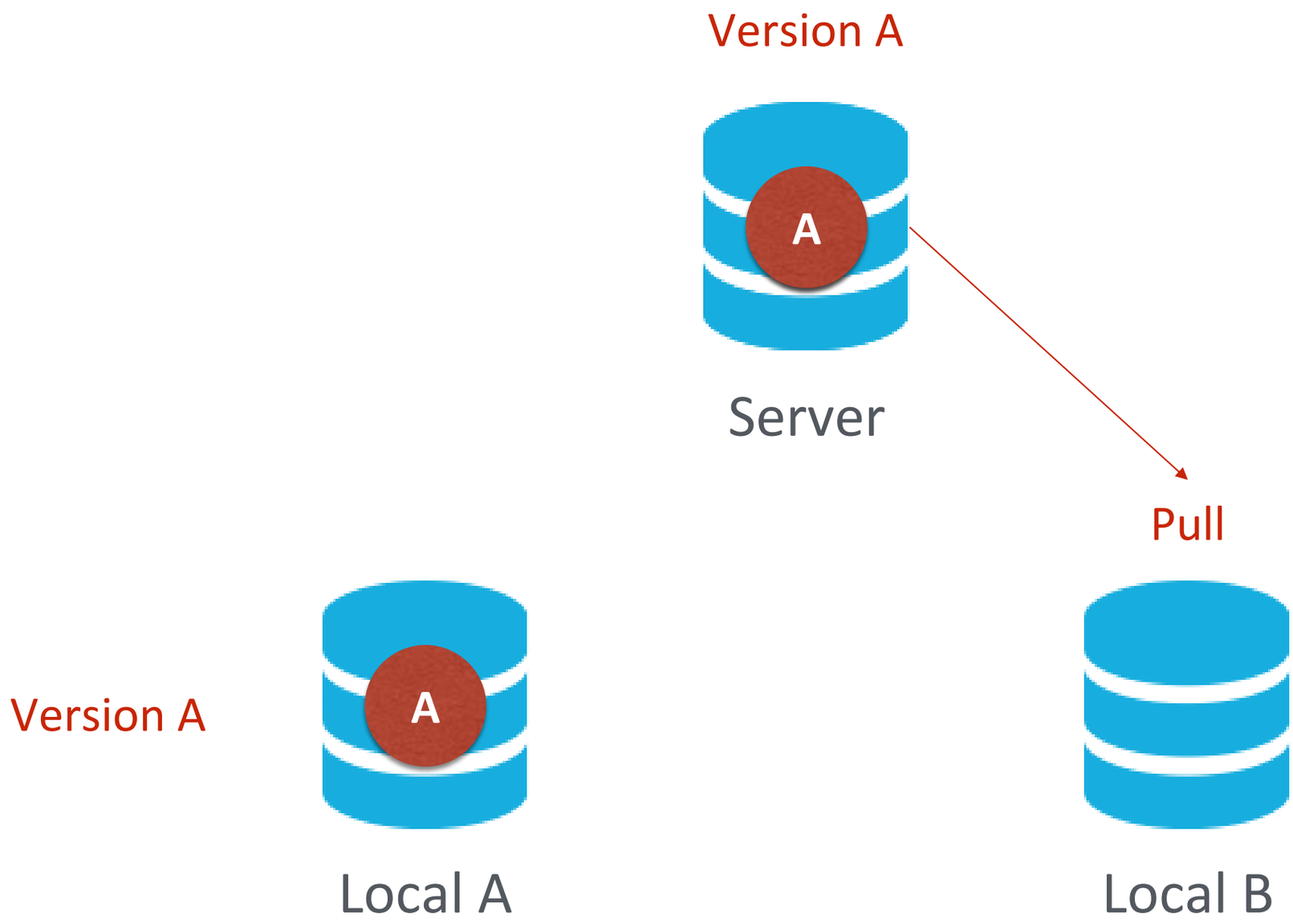
Local A

Local B

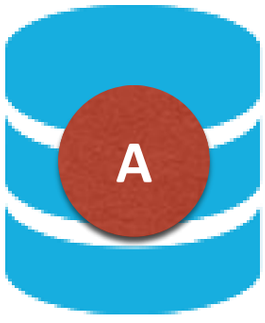
Collaboration Workflows (2/2)



Collaboration Workflows (2/2)



Collaboration Workflows (2/2)



Server

Pull

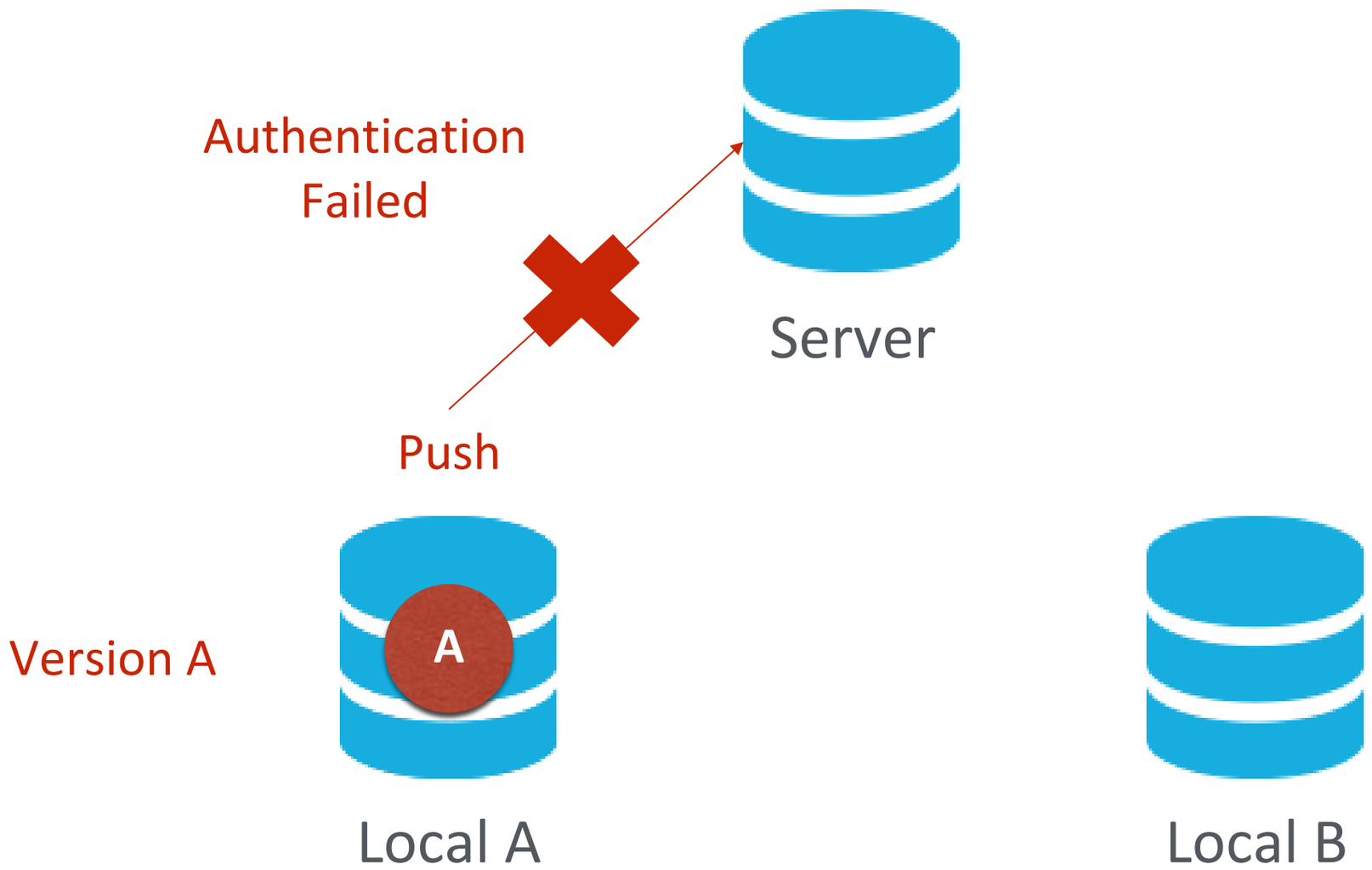
Local A

Local B

Version A

Something went wrong.

Authentication Failed



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

Author

Original Project

A

Forked Project

A2

Open a Merge Request

A

A1

A2

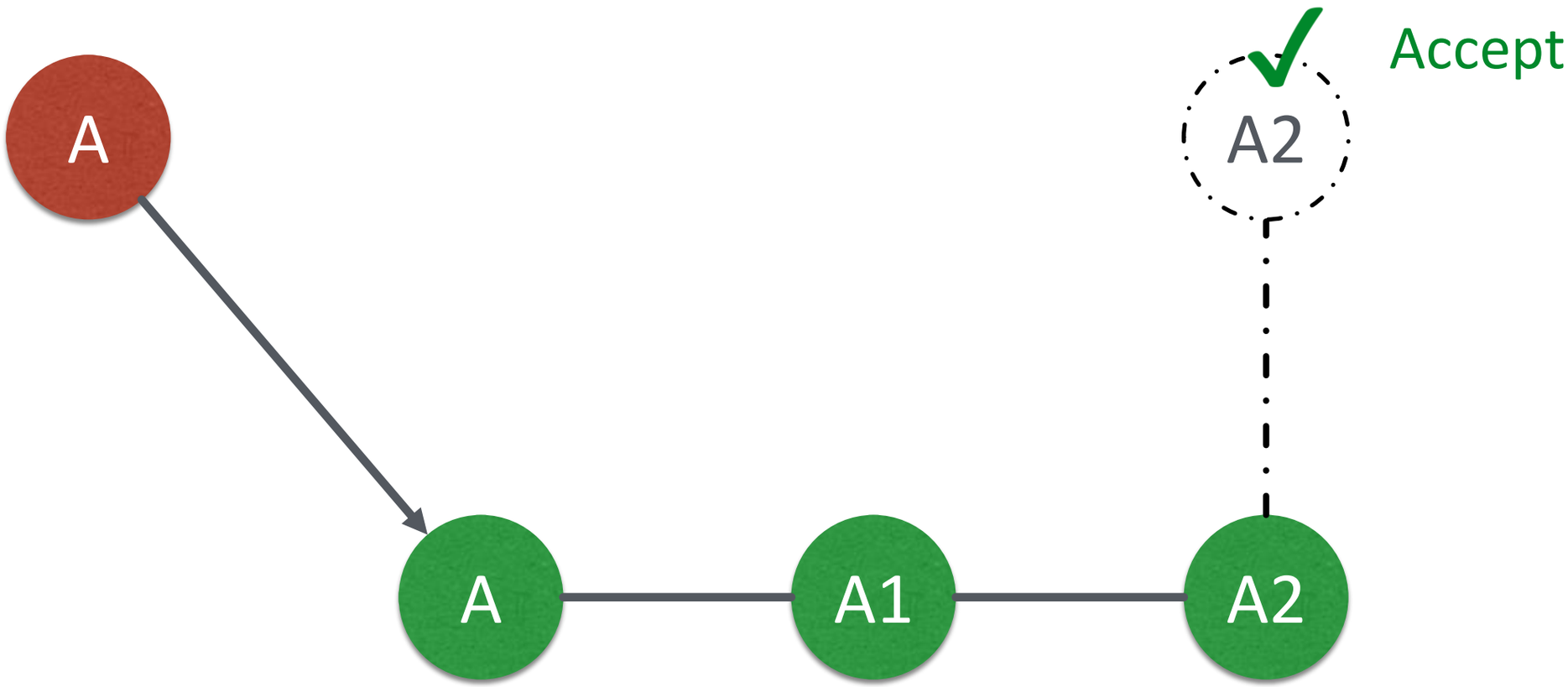
Forked from Red

Commit

Commit

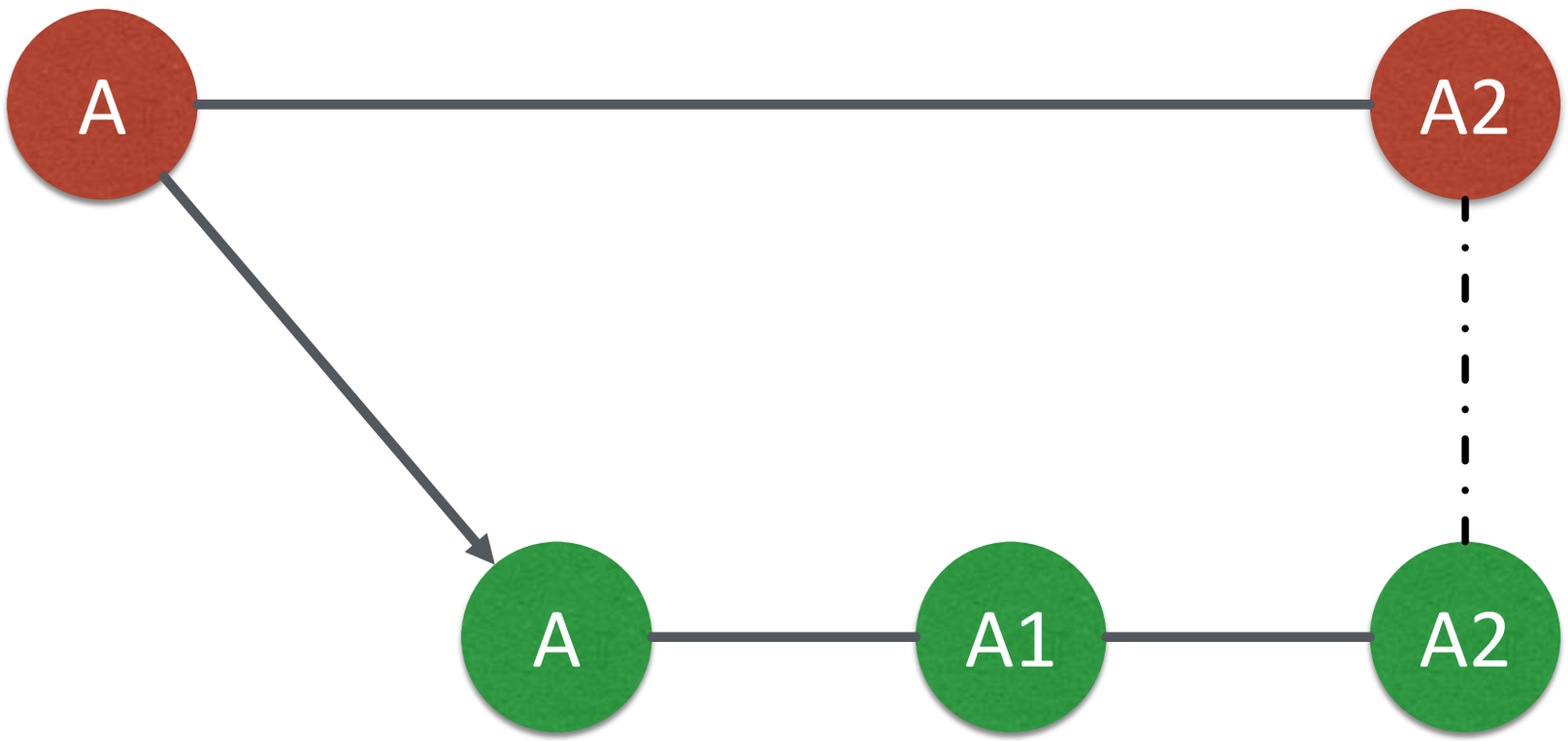
Editor

Author

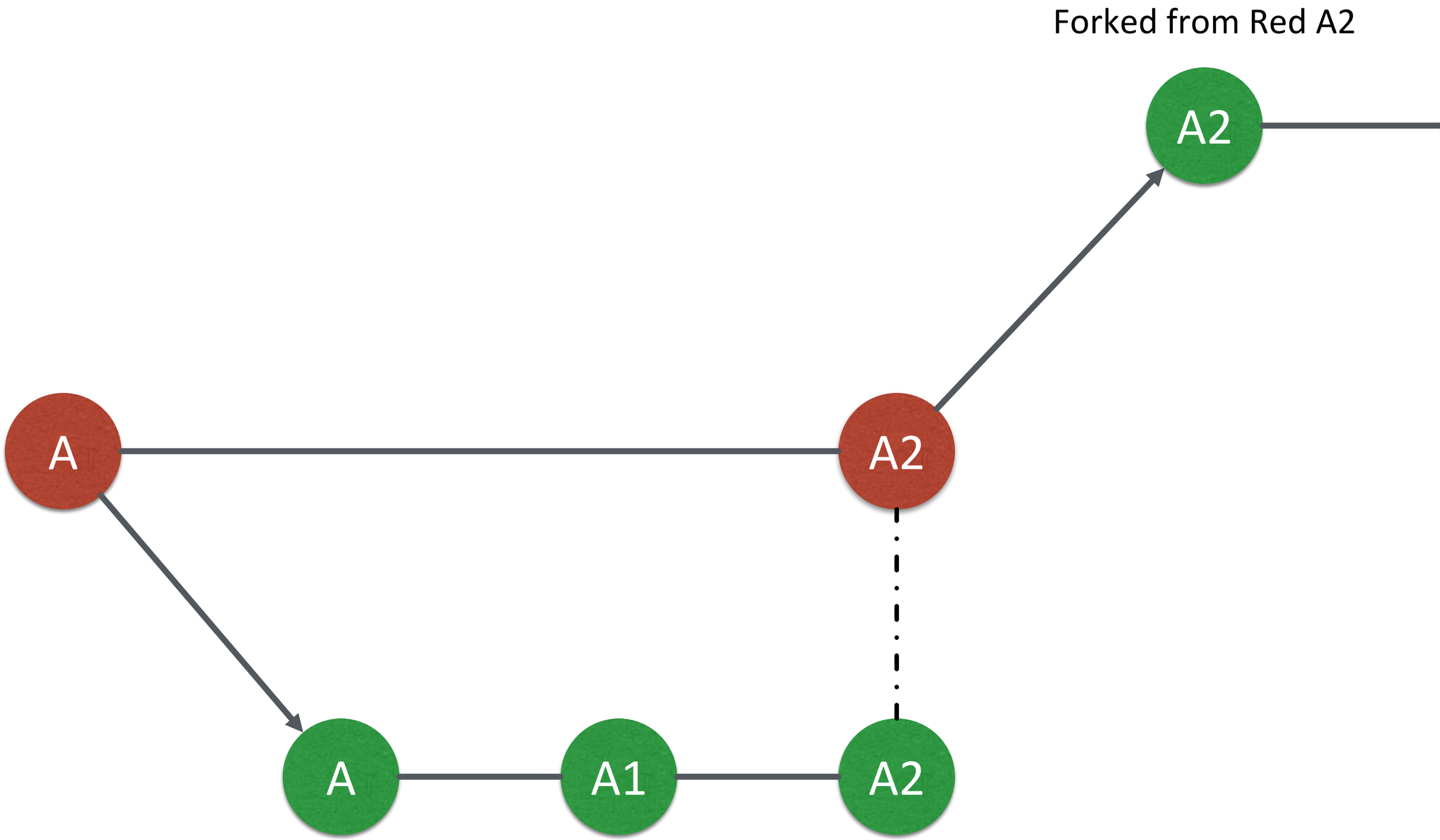


Editor

Author



Editor


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

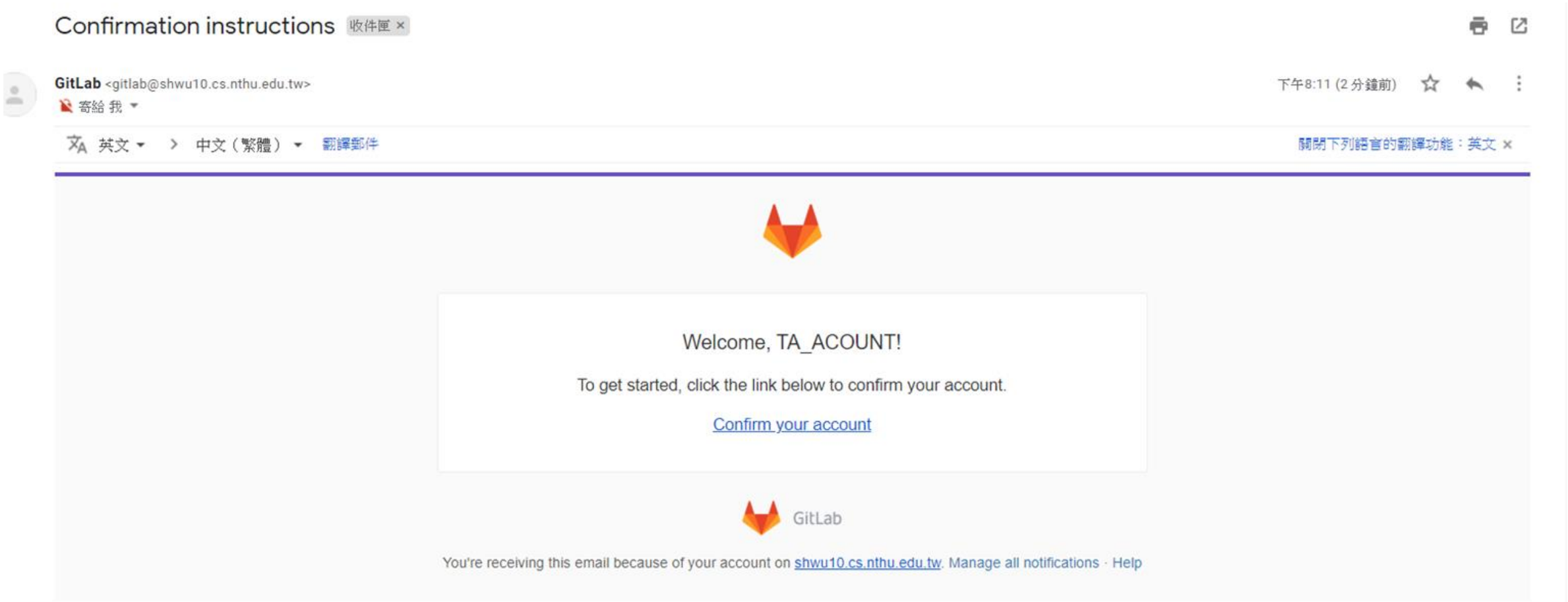
Gitlab

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

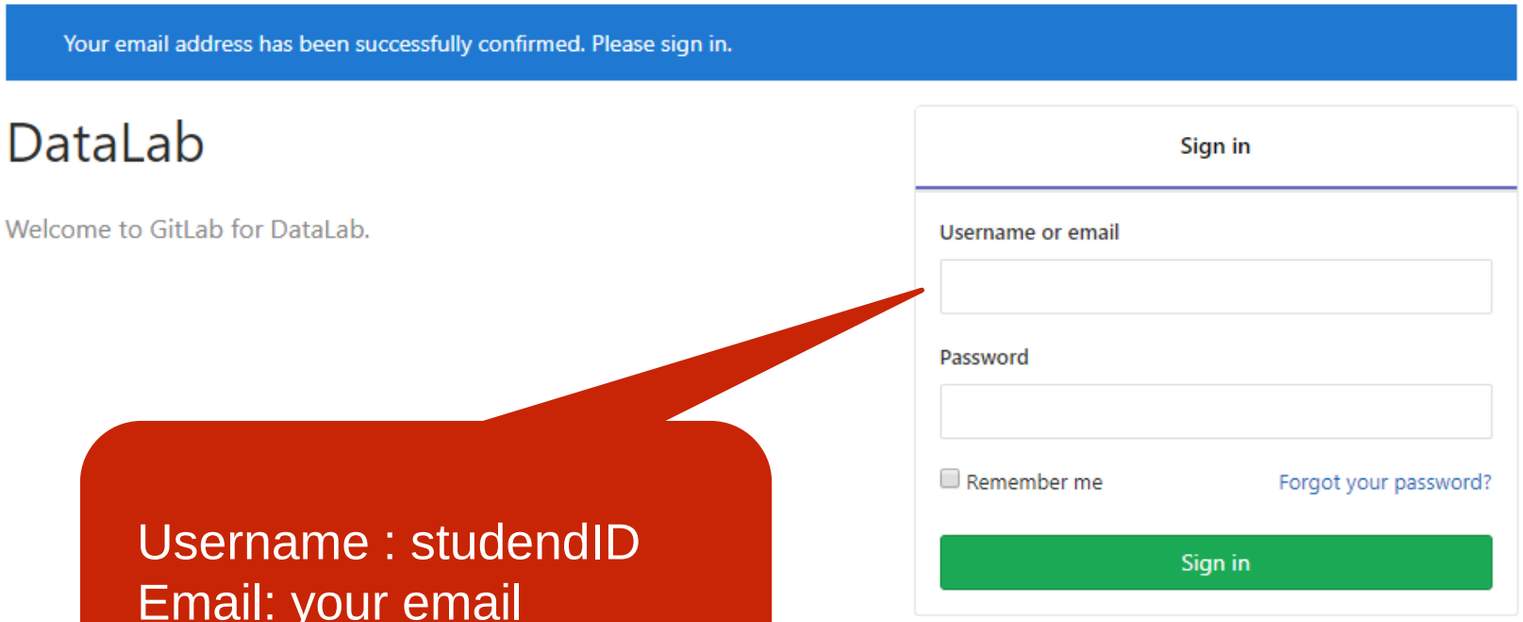
Gitlab

Gmail
搜尋郵件
撰寫
收件匣 9
已加星號
已延後
主要
社交網路 33 個新對話
Twitter
促銷內容
GitLab
Access to the courses / software-studio / 2020-sprin
GitLab
Confirmation instructions - Welcome, TA_ACOUNT! T

Gitlab


Confirmation instructions
收件匣 ×
GitLab <gitlab@shwu10.cs.nthu.edu.tw>
寄給 我
下午8:11 (2 分鐘前)
英文
中文（繁體）
翻譯郵件
關閉下列語言的翻譯功能：英文 ×
Welcome, TA_ACOUNT!
To get started, click the link below to confirm your account.
Confirm your account
GitLab
You're receiving this email because of your account on shwu10.cs.nthu.edu.tw. Manage all notifications · Help

Gitlab



Username : studendID
Email: your email
Password: studentID

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

courses › ••• › 2025-spring

2

2025-spring

Group ID: 6011

New subgroup | New project

Subgroups and projects | Shared projects | Archived projects

Search by name | Name

S **submission-exercise**

★ 0

1 hour ago

courses › ••• › 2025-spring › submission-exercise

S

submission-exercise

Project ID: 8735

Star 0 Fork 0

1. Click to fork

2 Commits 1 Branch 0 Tags 41 KB Files 41 KB Storage

master submission-exercise / + History Find file Web IDE Clone

Update README.md
Shao-Che Feng authored 1 hour ago 19a597ef

Upload File README Auto DevOps enabled Add LICENSE Add CHANGELOG Add CONTRIBUTING

Configure Integrations

Name	Last commit	Last update
README.md	Update README.md	1 hour ago

README.md

Lab 1 - 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 project.

Try it!

1. Fork this project.
2. Clone the **forked** project from Gitlab to your local environment.
3. Add a new file and write something.
4. Commit your work.
5. Push the repository to the server.

courses › ••• › 2024-spring › submission-exercise › **Fork project**

Fork project

A fork is a copy of a project.
Forking a repository allows you to make changes without affecting the original project.

Project name

submission-exercise

Project URL

https://shwu10.cs.nthu.edu.tw/ | Select a namespace

Project slug

submission-exercise

Want to house several dependent projects u

2. Select your name

Project description (optional)

Visibility level

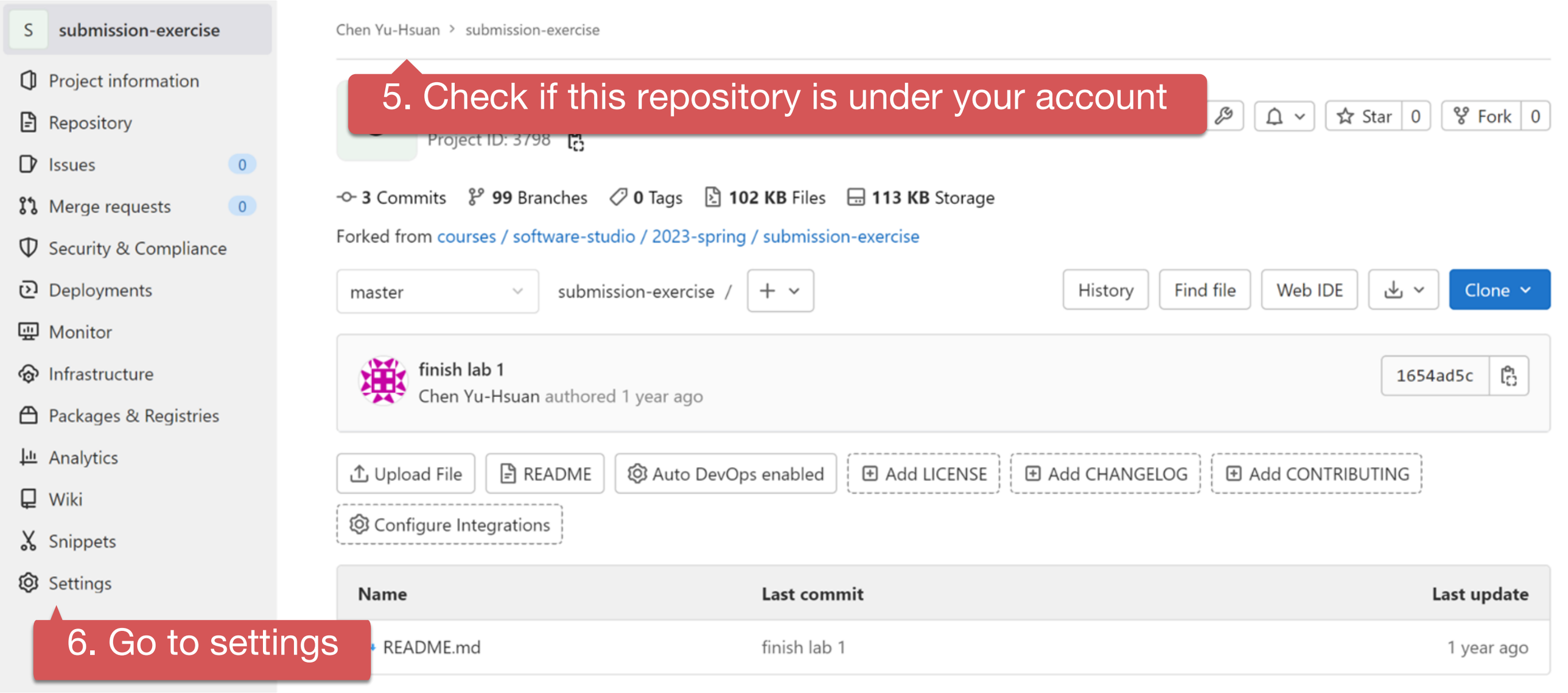
- Private
  Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.
- Inte
  The pro
- Publ
  The project can be accessed without any authentication.

3. Click Private

Fork project

Cancel

4. Click to fork


S submission-exercise
Project information
Repository
Issues 0
Merge requests 0
Security & Compliance
Deployments
Monitor
Infrastructure
Packages & Registries
Analytics
Wiki
Snippets
Settings
Chen Yu-Hsuan > submission-exercise
5. Check if this repository is under your account
Project ID: 3798
Star 0
Fork 0
3 Commits
99 Branches
0 Tags
102 KB Files
113 KB Storage
Forked from courses / software-studio / 2023-spring / submission-exercise
master
submission-exercise /
History
Find file
Web IDE
Clone
finish lab 1
Chen Yu-Hsuan authored 1 year ago
1654ad5c
Upload File
README
Auto DevOps enabled
Add LICENSE
Add CHANGELOG
Add CONTRIBUTING
Configure Integrations
Name
Last commit
Last update
README.md
finish lab 1
1 year ago
6. Go to settings

GitLab Menu Search GitLab

S submission-exercise

- Project information
- Repository
- Issues 0
- Merge requests 0
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Packages & Registries
- Analytics
- Wiki
- Snippets
- Settings
  - General
  - Integrations
  - Webhooks
  - Access Tokens
  - Repository
  - Monitor
  - Packages & Registries
  - Usage Quotas

Max file size is 200 KB.

Save changes

Visibility, project features, permissions

Collapse

Choose visibility level, enable/disable project features and their permissions, disable email notifications, and show default award emoji.

7. Set project to private

Project visibility

Manage who can see the project in the public access directory. Learn more.

Private

The project is accessible only by members of the project. Access must be granted explicitly to each user.

Issues

Flexible tool to collaboratively develop ideas and plan work in this project. Learn more.

Only Project Members

Repository

View and edit files in this project.

Only Project Members

Merge requests

Submit changes to be merged upstream.

Only Project Members

Forks

Users can copy the repository to a new project.

Only Project Members

GitLab Menu Search GitLab

S submission-exercise

- Project information
- Repository
- Issues 0
- Merge requests 0
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Packages & Registries
- Analytics
- Wiki
- Snippets
- Settings
  - General
  - Integrations
  - Webhooks

Only Project Members

Disable email notifications

Override user notification preferences for all project members.

Show default award emojis

Always show thumbs-up and thumbs-down award emoji buttons on issues, merge requests, and snippets.

War

Hig languages, but can also be used in potential e oits.

8. Scroll down and save changes

Save changes

Merge requests

Expand

Choose your merge method, merge options, merge checks, and merge suggestions.

Badges

Expand

Customize this project's badges. What are badges?

Service Desk

Expand

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

courses › ••• › 2025-spring › submission-exercise

S

submission-exercise

Project ID: 6444

Star 0 Fork 0

0 Commits 1 Branch 0 Tags 20 KB Files 20 KB Storage

master submission-exercise / + History Find file Web IDE Clone

Initial commit
Chen Yu-Hsuan authored 3 days ago

Upload File Auto DevOps enabled Add README Add LICENSE Add

Configure Integrations

Clone with SSH

git@shwu10.cs.nthu.edu.tw:

Clone with HTTPS

https://shwu10.cs.nthu.edu.tw/cour

2. Copy the link

1. Choose HTTPS

Visual Studio Code (SSH)

Visual Studio Code (HTTPS)

Name	Last commit
README.md	Initial commit

README.md

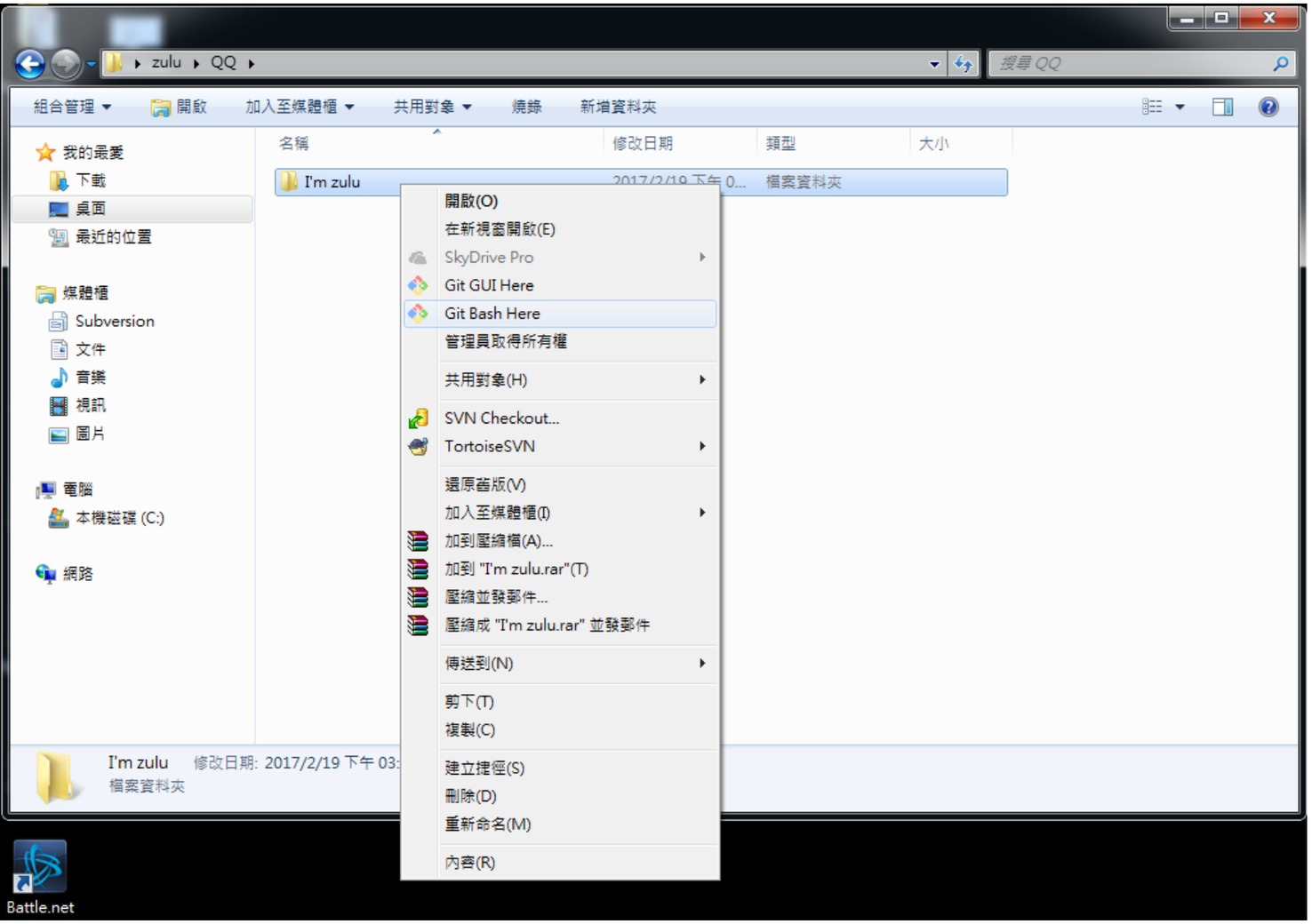
Lab 1 - Practice Submission

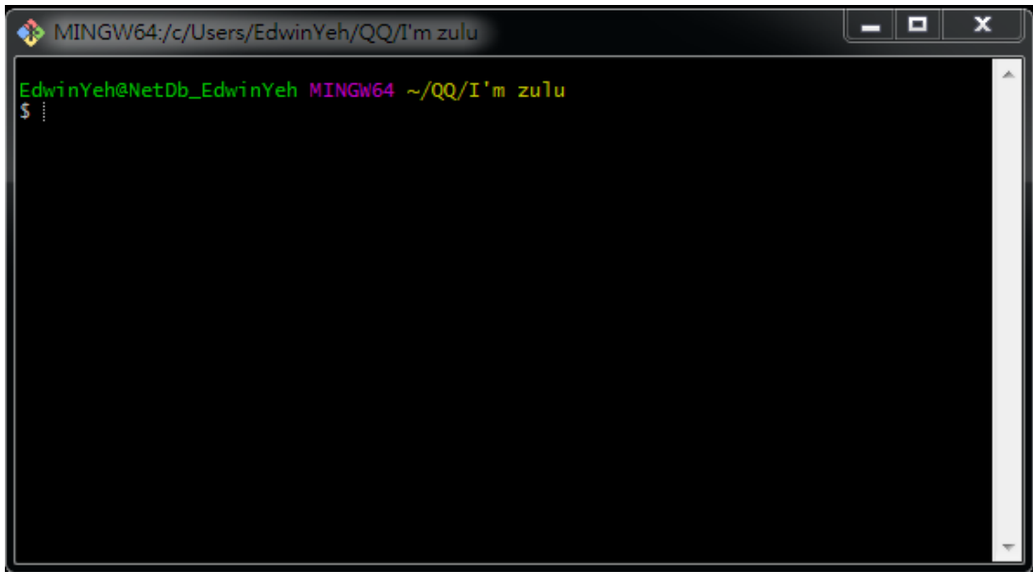
This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting a lab or project.

Try it!

1. Fork this project.
2. Clone the **forked** project from Gitlab to your local environment.
3. Add a new file and write something.
4. Commit your work.
5. Push the repository to the server.
6. Send a merge request of **your branch** to the origin repository.

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 wit

    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

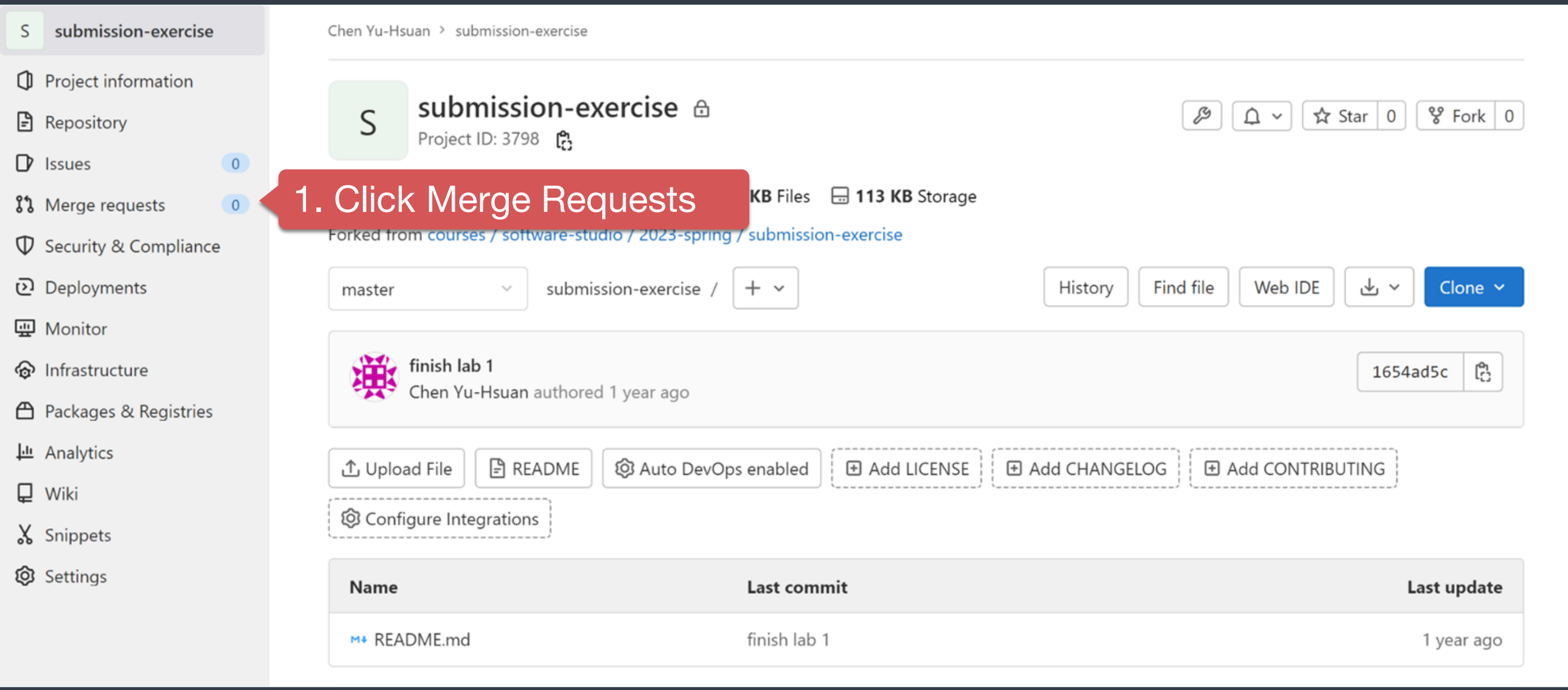
```
~/SS-Projects/submission-exercise  master  git push -u origin master
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 497 bytes | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
To https://shwu10.cs.nthu.edu.tw/ss-student/submission-exercise.git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


S submission-exercise
Project information
Repository
Issues 0
Merge requests 0
Security & Compliance
Deployments
Monitor
Infrastructure
Packages & Registries
Analytics
Wiki
Snippets
Settings
Chen Yu-Hsuan > submission-exercise
S
submission-exercise
Project ID: 3798
Star 0
Fork 0
1. Click Merge Requests
KB Files
113 KB Storage
Forked from courses / software-studio / 2023-spring / submission-exercise
master
submission-exercise /
History
Find file
Web IDE
Clone
finish lab 1
Chen Yu-Hsuan authored 1 year ago
1654ad5c
Upload File
README
Auto DevOps enabled
Add LICENSE
Add CHANGELOG
Add CONTRIBUTING
Configure Integrations
Name
Last commit
Last update
README.md
finish lab 1
1 year ago

GitLab Projects Groups Activity Milestones Snippets This project Search

S submission-exercise

Project
Repository
Issues 0
Merge Requests 0
Wiki
Snippets
Settings

Collapse sidebar

Merge requests are a place to propose changes you've made to a project and discuss those changes with others

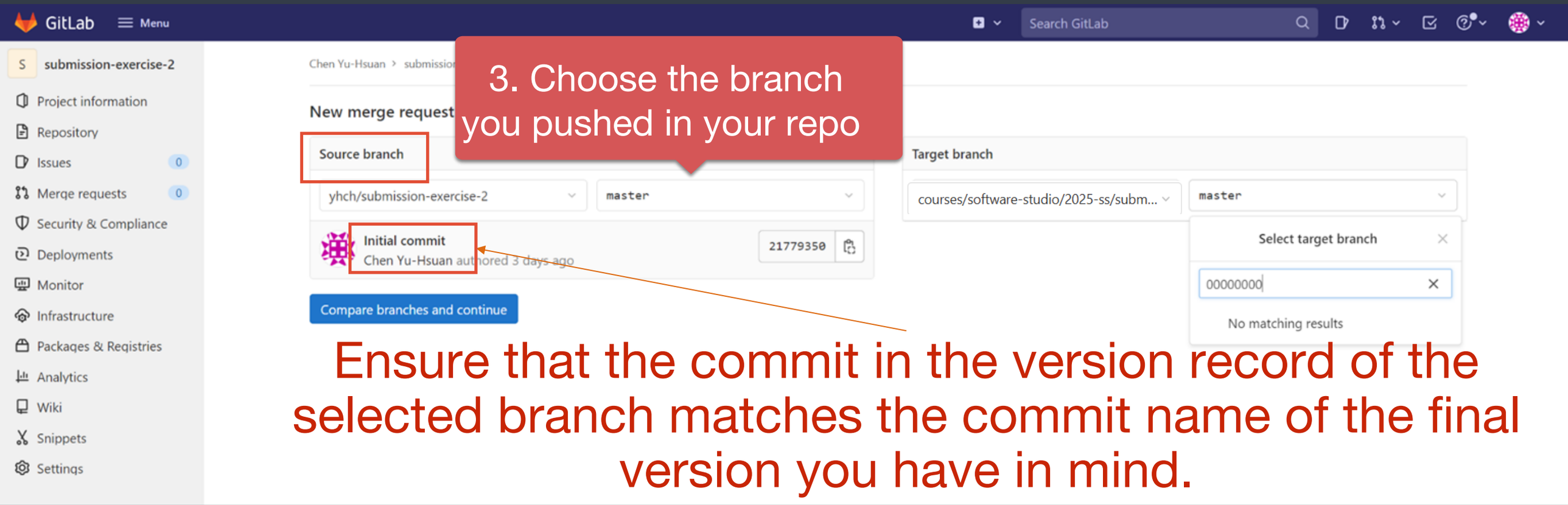
Interested parties can even contribute by pushing commits if they want to.

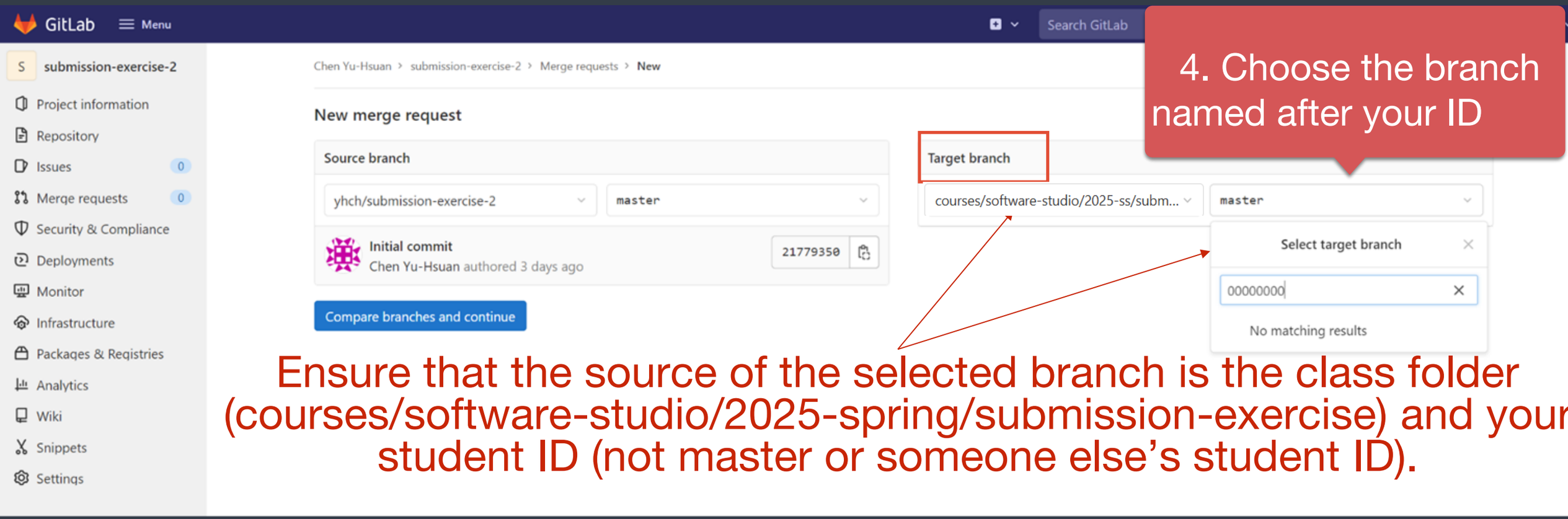
New merge request

New merge request

2. New merge request

https://shwu10.cs.nthu.edu.tw/103060010/submission-exercise/merge_requests/new


GitLab
Menu
Search GitLab
submission-exercise-2
Project information
Repository
Issues 0
Merge requests 0
Security & Compliance
Deployments
Monitor
Infrastructure
Packages & Registries
Analytics
Wiki
Snippets
Settings
Chen Yu-Hsuan › submissio
New merge request
3. Choose the branch you pushed in your repo
Source branch
yhch/submission-exercise-2
master
Initial commit
Chen Yu-Hsuan authored 3 days ago
21779350
Compare branches and continue
Target branch
courses/software-studio/2025-ss/subm...
master
Select target branch
00000000
No matching results
Ensure that the commit in the version record of the selected branch matches the commit name of the final version you have in mind.


GitLab
Menu
Search GitLab
submission-exercise-2
Project information
Repository
Issues 0
Merge requests 0
Security & Compliance
Deployments
Monitor
Infrastructure
Packages & Registries
Analytics
Wiki
Snippets
Settings
Chen Yu-Hsuan › submission-exercise-2 › Merge requests › New
New merge request
Source branch
yhch/submission-exercise-2
master
Initial commit
Chen Yu-Hsuan authored 3 days ago
21779350
Compare branches and continue
Target branch
courses/software-studio/2025-ss/subm...
master
Select target branch
00000000
No matching results
4. Choose the branch named after your ID
Ensure that the source of the selected branch is the class folder (courses/software-studio/2025-spring/submission-exercise) and your student ID (not master or someone else's student ID).

GitLab Menu Search GitLab

S submission-exercise-2

- Project information
- Repository
- Issues 0
- Merge requests 0
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Packages & Registries
- Analytics
- Wiki
- Snippets
- Settings

Chen Yu-Hsuan › submission-exercise-2 › Merge requests › New

New merge request

Source branch

yhch/submission-exercise-2 | master

Initial commit
Chen Yu-Hsuan authored 3 days ago 21779350

Target branch

courses/software-studio/2025-ss/subm... | master

Select target branch

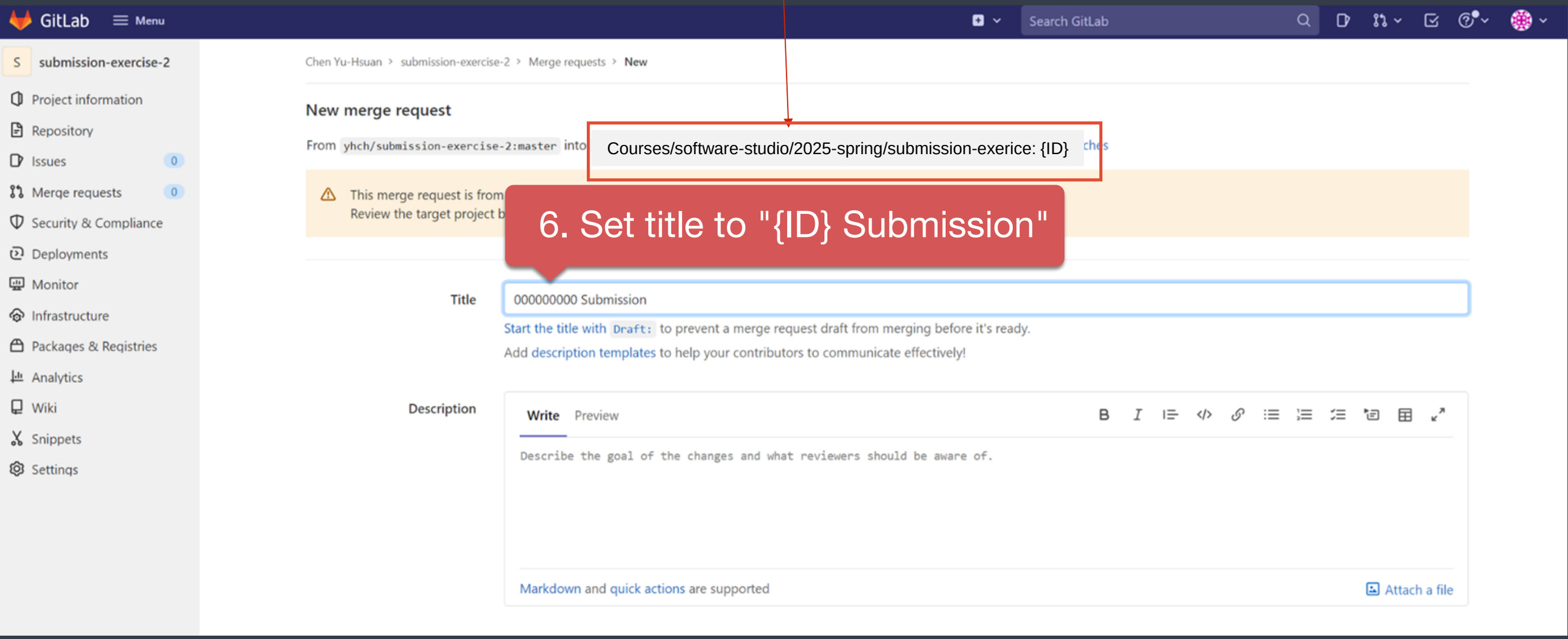
00000000

No matching results

Compare branches and continue

5. Compare branches

You can also confirm or change whether the selected branch is correct in this section. For instance, in the picture, the course branch selected is master, which is incorrect. The correct one should be courses/software-studio/2025-spring/submission-exercise: {your student ID}.



Description

Write Preview

Describe the goal of the changes and what reviewers should be aware of.

Markdown and quick actions are supported

Attach a file

Assignee Unassigned Assign to me

Reviewer Unassigned

Milestone Milestone

Labels Labels

Merge options Squash commits when merge request is accepted.

Contribution Allow commits from members who can merge to the target branch. About this feature

Not available for private projects

Create merge request

7. If everything is OK, submit your merge request

Notice - Don’t do this

Notice - Don’t do this

Notice - Don’t do this

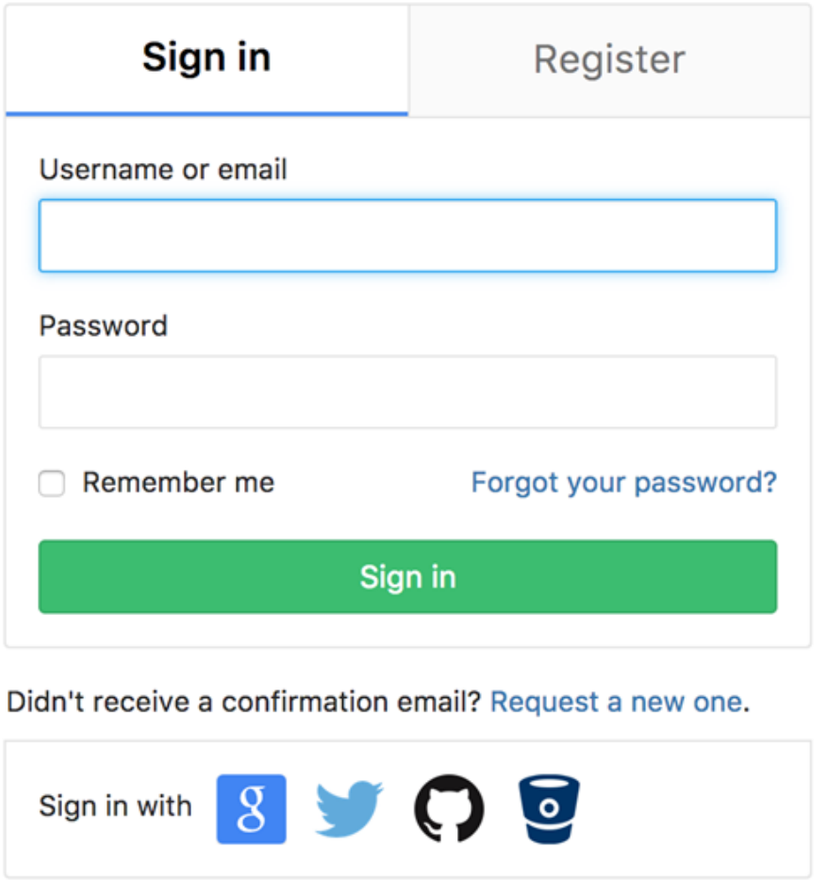
GitLab.com

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

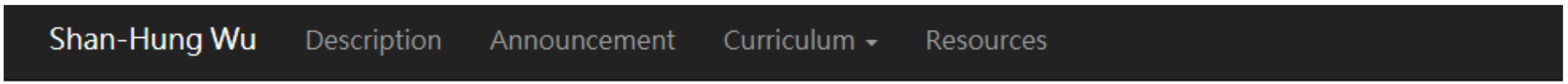
- **Explore projects on GitLab.com** (no login needed)
- **More information about GitLab.com**
- **GitLab.com Support Forum**

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

- **Privacy policy**
- **GitLab.com Terms**.



The GitLab setup in our laboratory is accessible through the link.



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.

GitLab iLMS System

Here!!!!!

Outline

- General Rule
- Introduction to Git
- Version control
- Branch and merge
- 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

```
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>';
```

0 18 Ln 38, Col 72 UTF-8 CRLF JavaScript

ATOM
A hackable text editor
for the 21st Century
build
docs
dot-atom
exports
keymaps
menus
node_modules
resources
script
spec
src
static
atom.coffee
Settings
Essential: Atom global for dealing with packages, themes, menus, and the wir
#
An instance of this class is always available as the `atom` global.
module.exports =
class Atom extends Model
@version: 1 # Increment this when the serialization format changes
Load or create the Atom environment in the given mode.
#
Returns an Atom instance, fully initialized.
@loadOrCreate: (mode) ->
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 “Git Command-line Tool Installation”.
- Try to submit in GitLab.