

Introduction to Git

Database Systems
DataLab, CS, NTHU
Spring, 2022

Outline

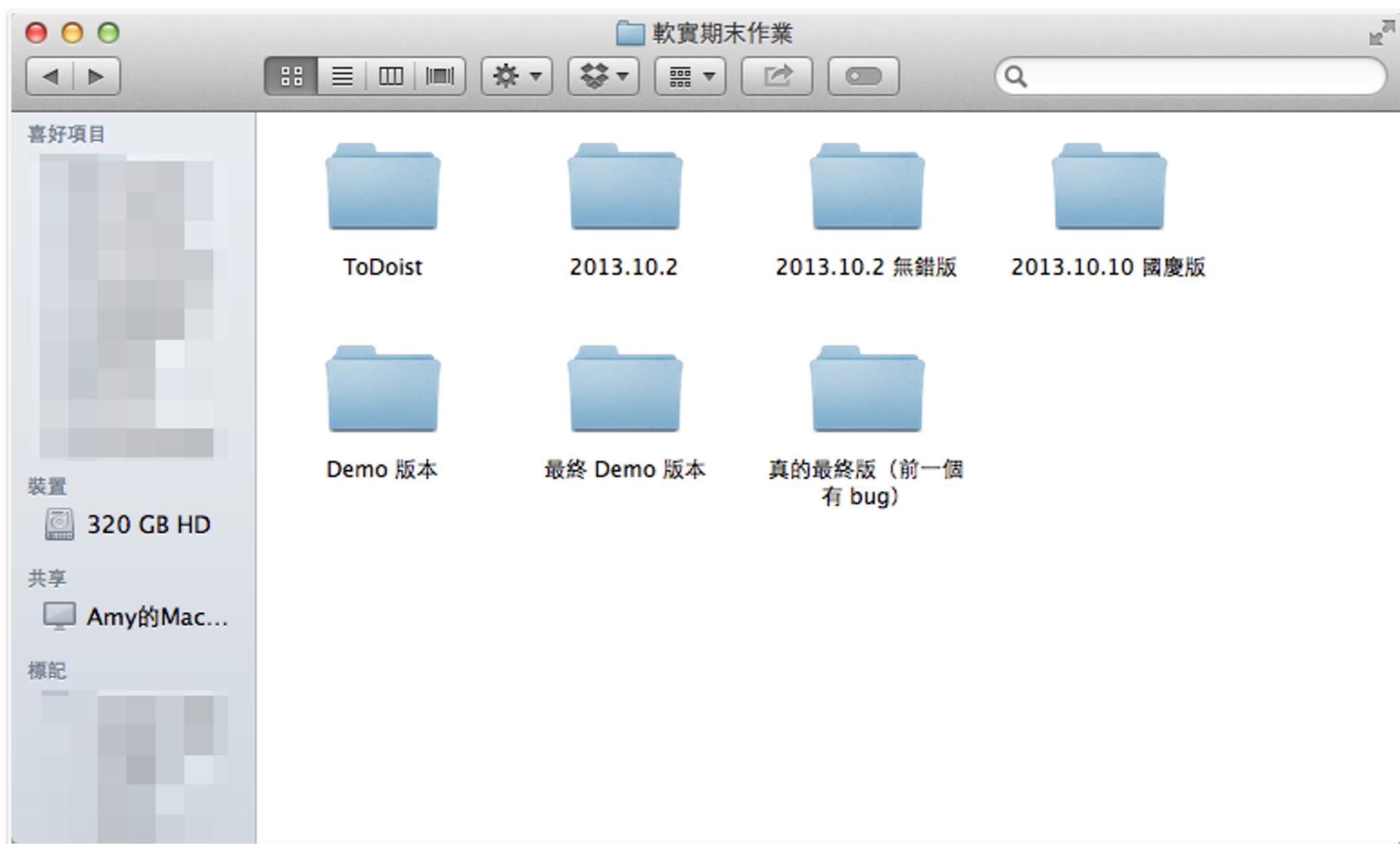
- Version control system
- Git basics
- Git branch
- Remote repository

Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Why Version Control ?

Students' VCS



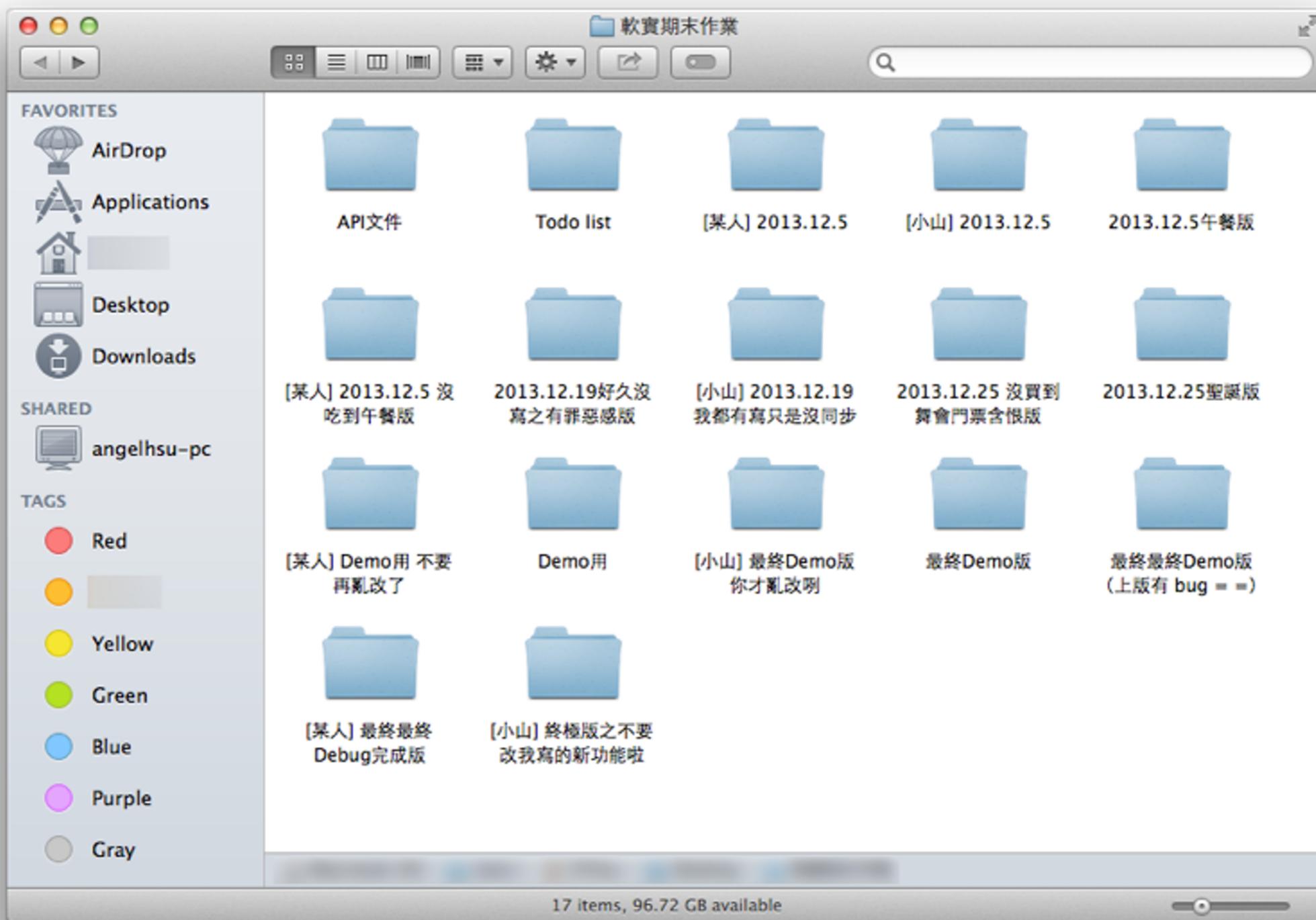
How to work with others?



?



Dropbox VCS in Reality



Version Control System

- Store the projects, keep your **revision history**
- **Synchronization** between modifications made by different developers

NOW GIT!!!



memecrunch.com

<https://memecrunch.com/meme/8QNIV/now-git>

Outline

- Version control system
- **Git basics**
- Git branch
- Remote repository

Git

- Git is a version control system which is
 - Fast
 - Easy to use
 - Distributed
 - Able to handle large project
(ex. Linux Kernel 27.8 million lines)
- A git repository is a mini database that tracks your files

Installation

- Please check this link
 - <http://git-scm.com/book/en/Getting-Started-Installing-Git>

Configuration

- Modify `~/.gitconfig`
- Or, type in following commands

```
git config --global user.name "your name"
```

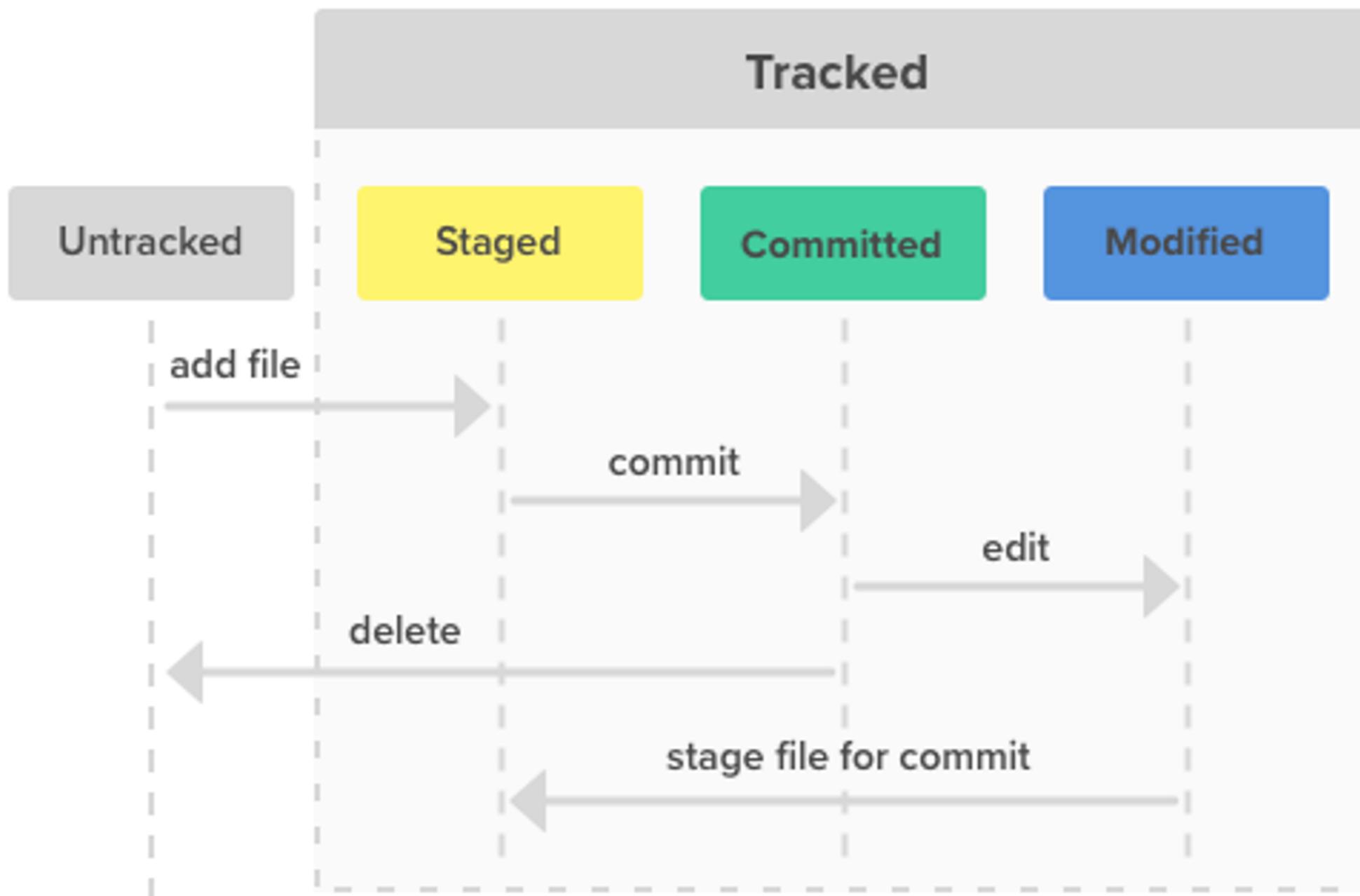
```
git config --global user.email "your@email.com"
```

- * For more information, please refer this [link](#).

Creating a new Repository

- Two ways to create a repository
 - Initializing a Repository in an Existing Directory
`git init`
 - Cloning an Existing Repository
 - We will talk about it later
 - The repository information will be stored in the **.git** directory

Committing A Version



Committing A Version

- Staging (adding) a file

`git add [file name]`

- Staging all files in the current directory

`git add -A`

- Committing

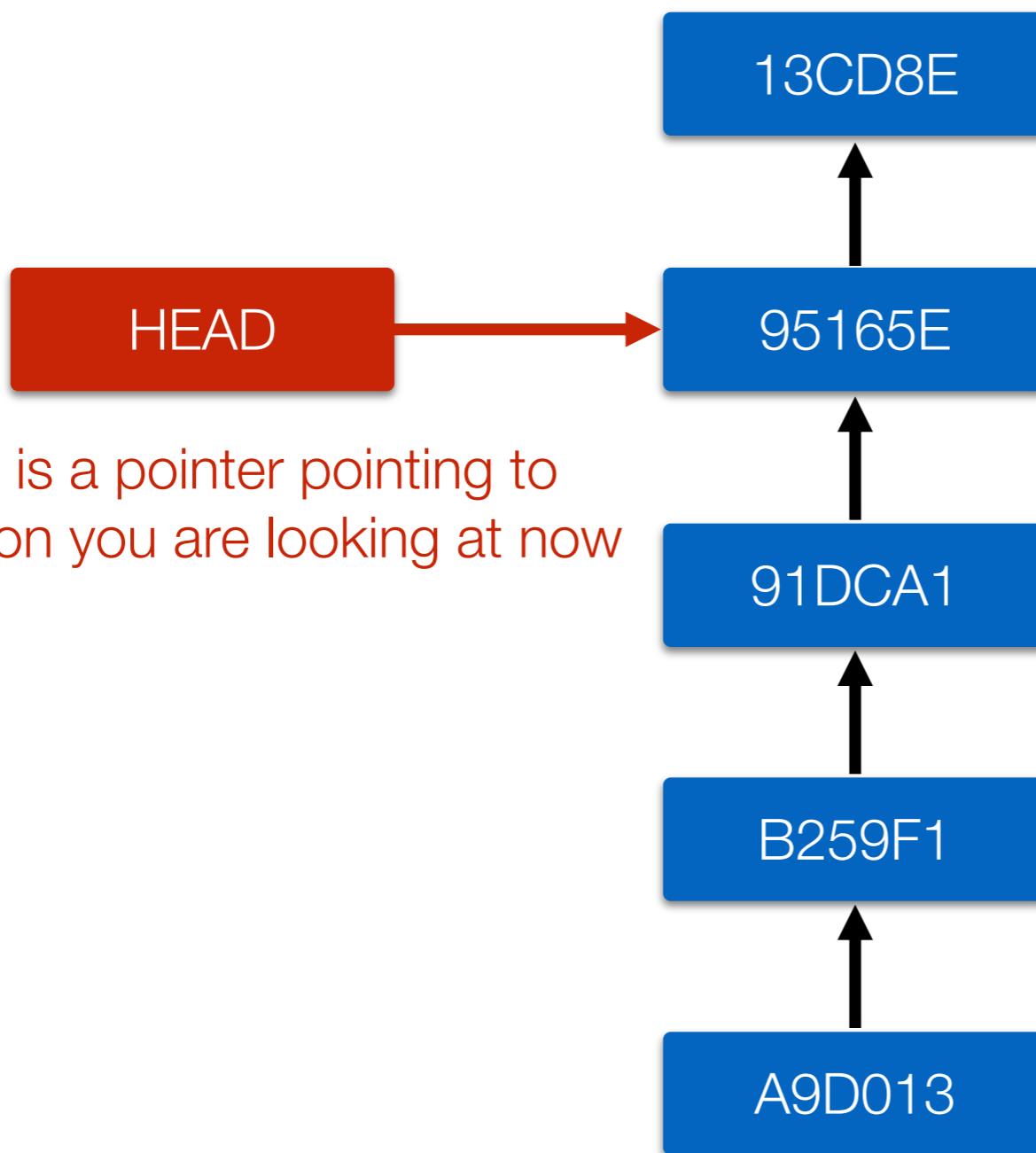
`git commit -m “[message]”`

Status

- Checking the current status and the current branch

`git status`

A History Tree



HEAD is a pointer pointing to the version you are looking at now

The versions you committed form a history tree

Logs

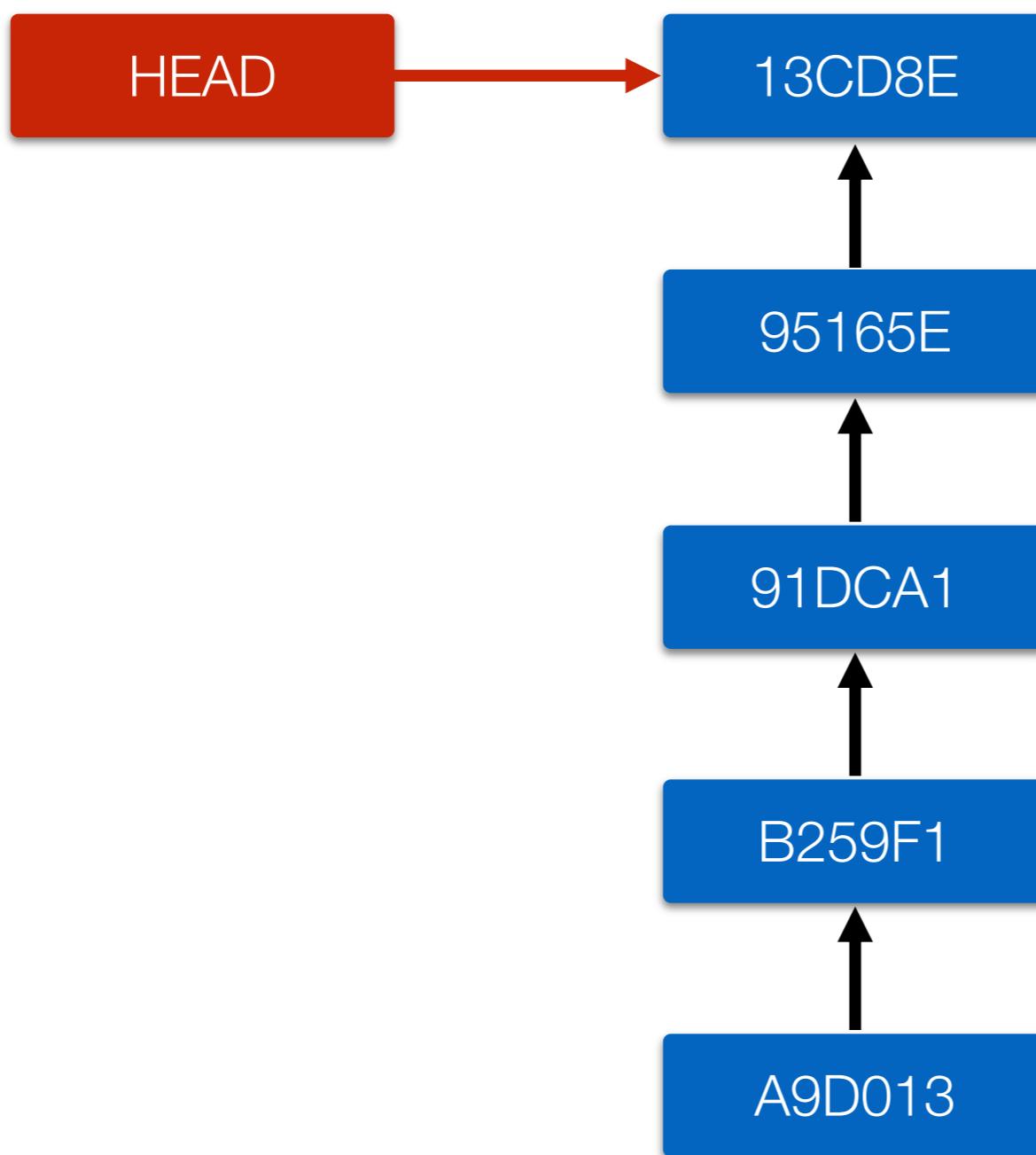
- Listing the log
`git log`
- Listing each log in one line
`git log --oneline`

```
user@DESKTOP-UUF072K MINGW64 ~/Desktop/test/practice-submission (master)
$ git log
commit 25f75ca42a96a900a6f87994a6b4fd70583d5561 (HEAD -> master, origin/master, origin/HEAD)
Author: kkeevin123456 <kkeevin123456@gmail.com>
Date:   Sat Mar 6 11:02:23 2021 +0800

    add branch

commit d7533bf6016e3faf834e9e38f8dfd552960728d1 (origin/p123786579, origin/109065541, origin/109062608, origin/
, origin/107062301, origin/107062261, origin/107062233, origin/107062231, origin/107062230, origin/107062228,
062123, origin/107062115, origin/107060023, origin/107060015, origin/107060008, origin/107060007, origin/107034
in/107000115, origin/107000105, origin/106072108, origin/106070034, origin/106070033, origin/106070007, origin/
, origin/106062133, origin/106062110, origin/106062109, origin/106033233, origin/106021225, origin/106012042,
061701, origin/101000030, p123786579, 109065541, 109062608, 108065702, 108003809, 107062381, 107062325, 1070623
82, 107062163, 107062140, 107062134, 107062123, 107062115, 107060023, 107060015, 107060008, 107060007, 10703405
4, 106070033, 106070007, 106062315, 106062306, 106062214, 106062205, 106062142, 106062137, 106062133, 106062110
, 104061701, 101000030)
Author: chanchishen <ccsch@datalab.cs.nthu.edu.tw>
```

Checking Out A Version

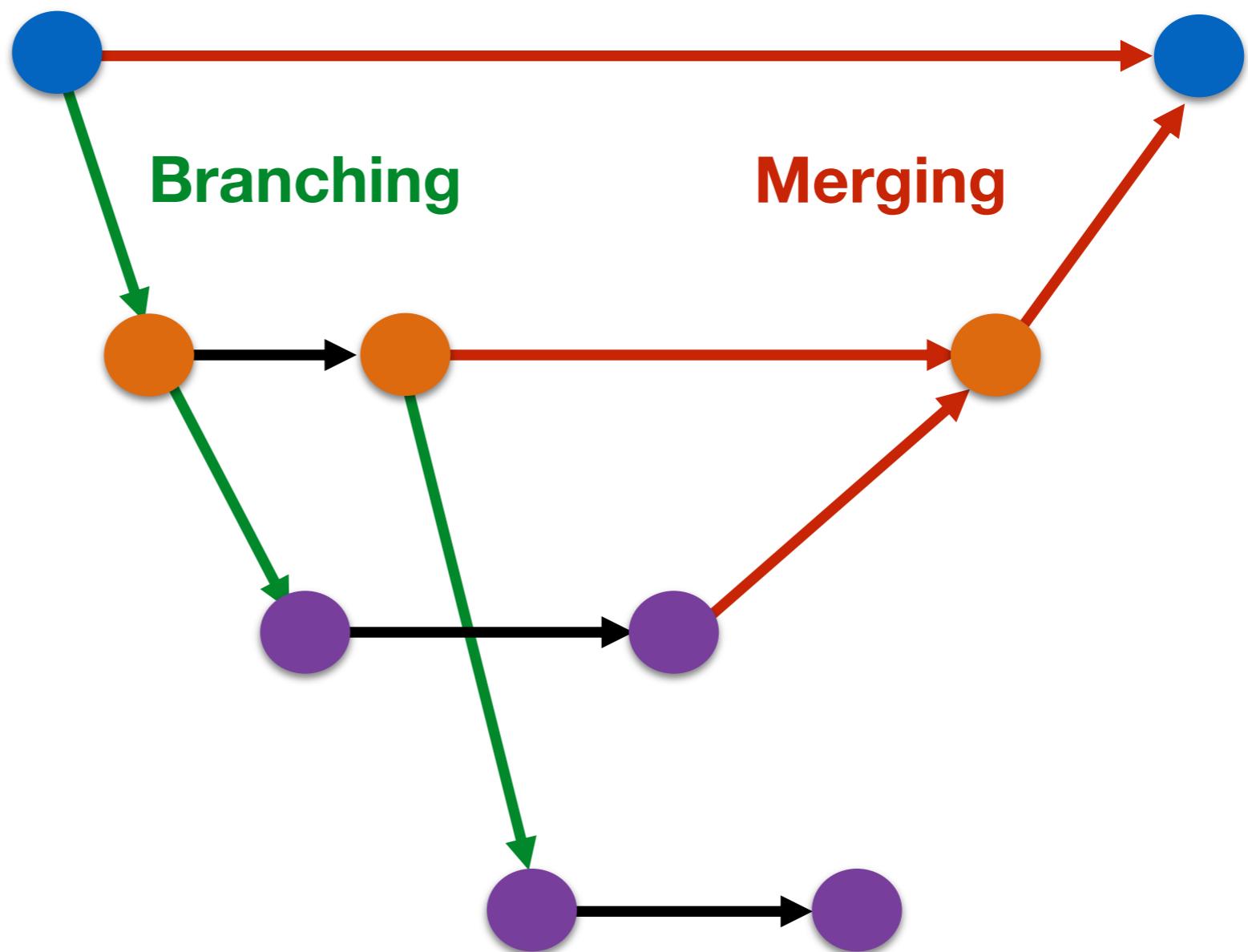
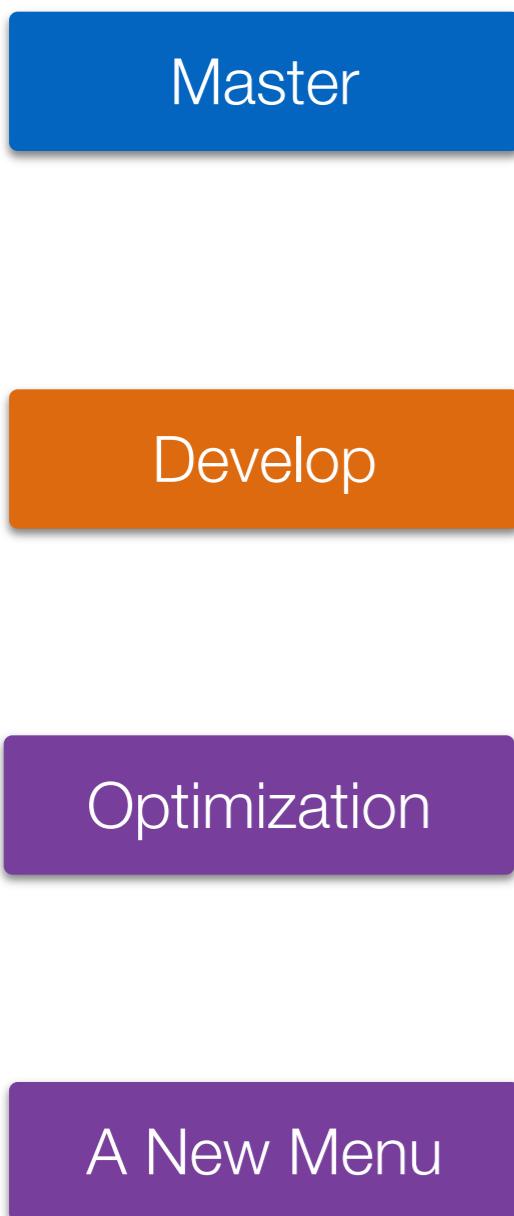


`git checkout 91DCA1`

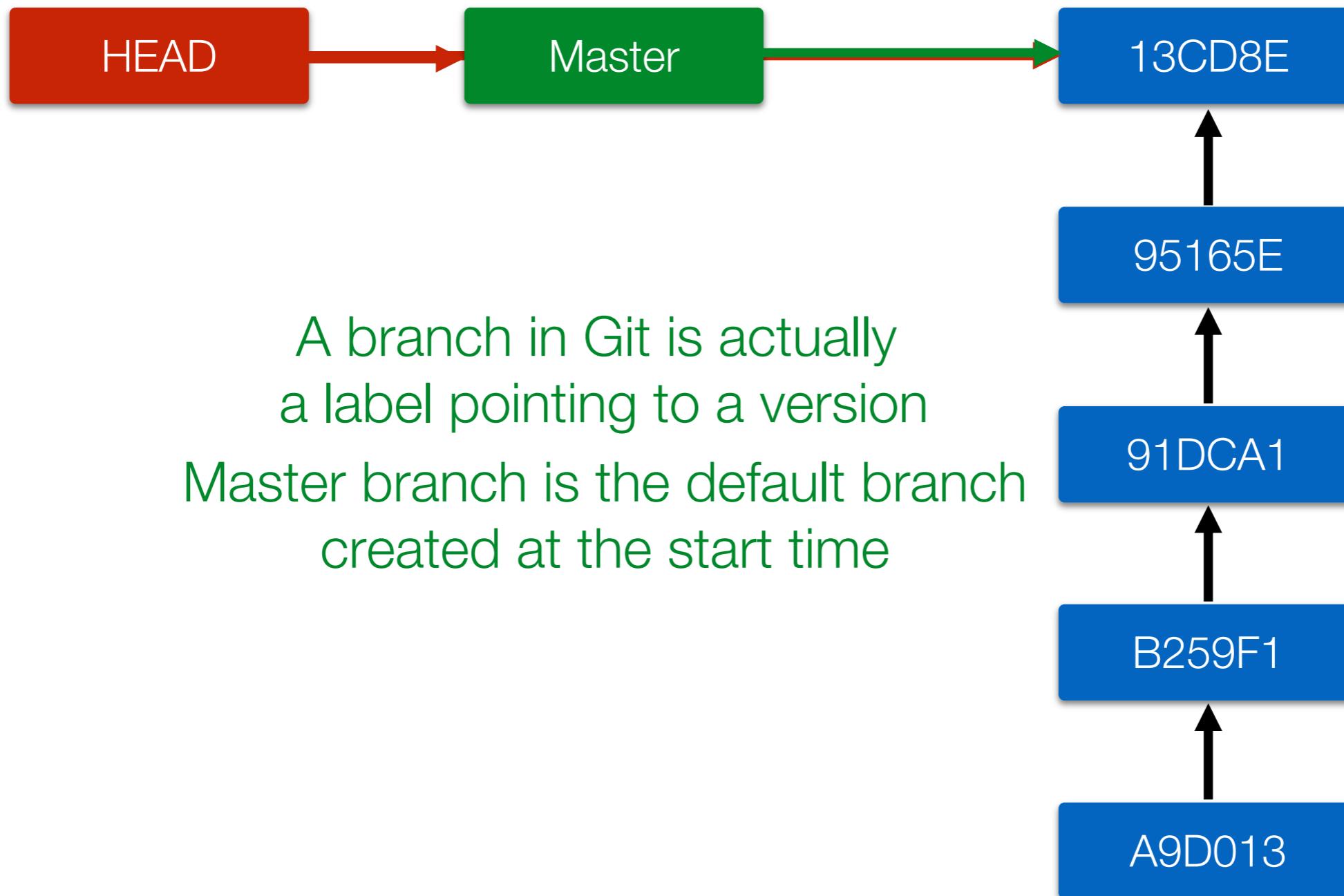
Outline

- Version control system
- Git basics
- **Git branch**
- Remote repository

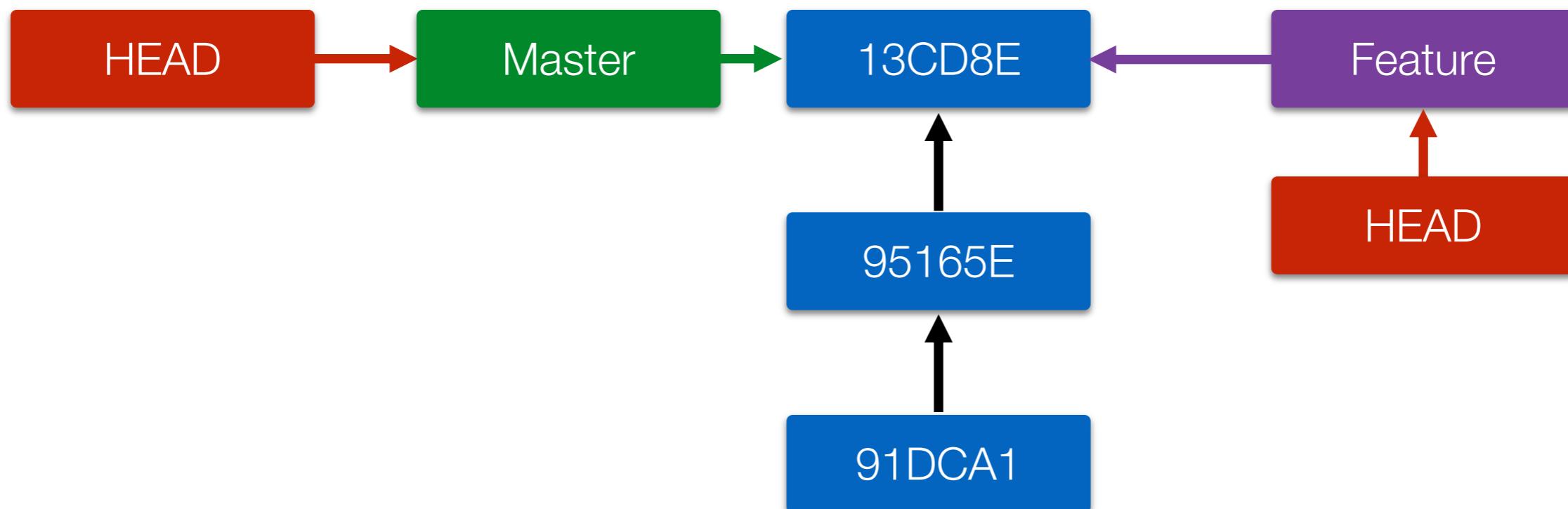
Branches



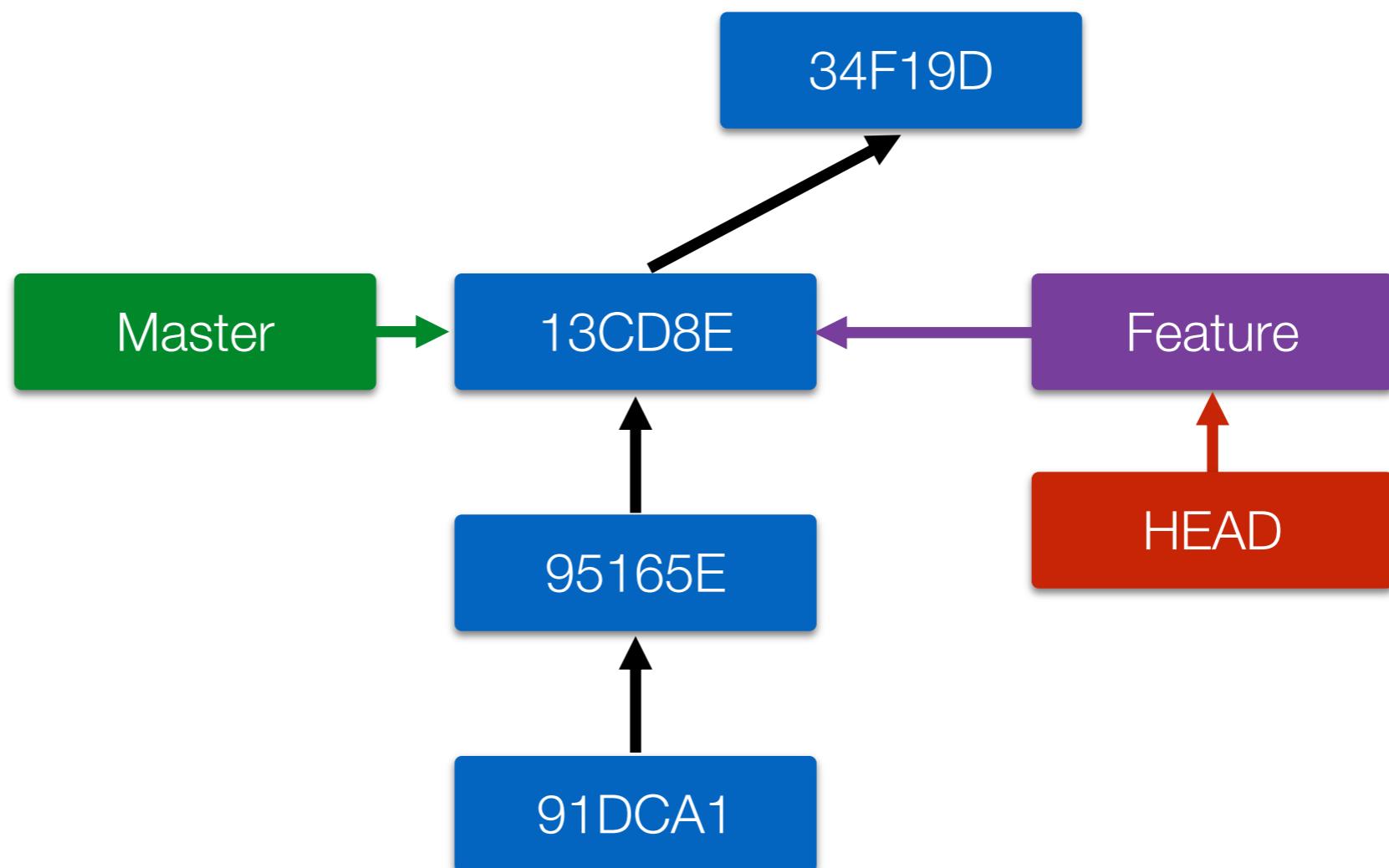
The Master Branch



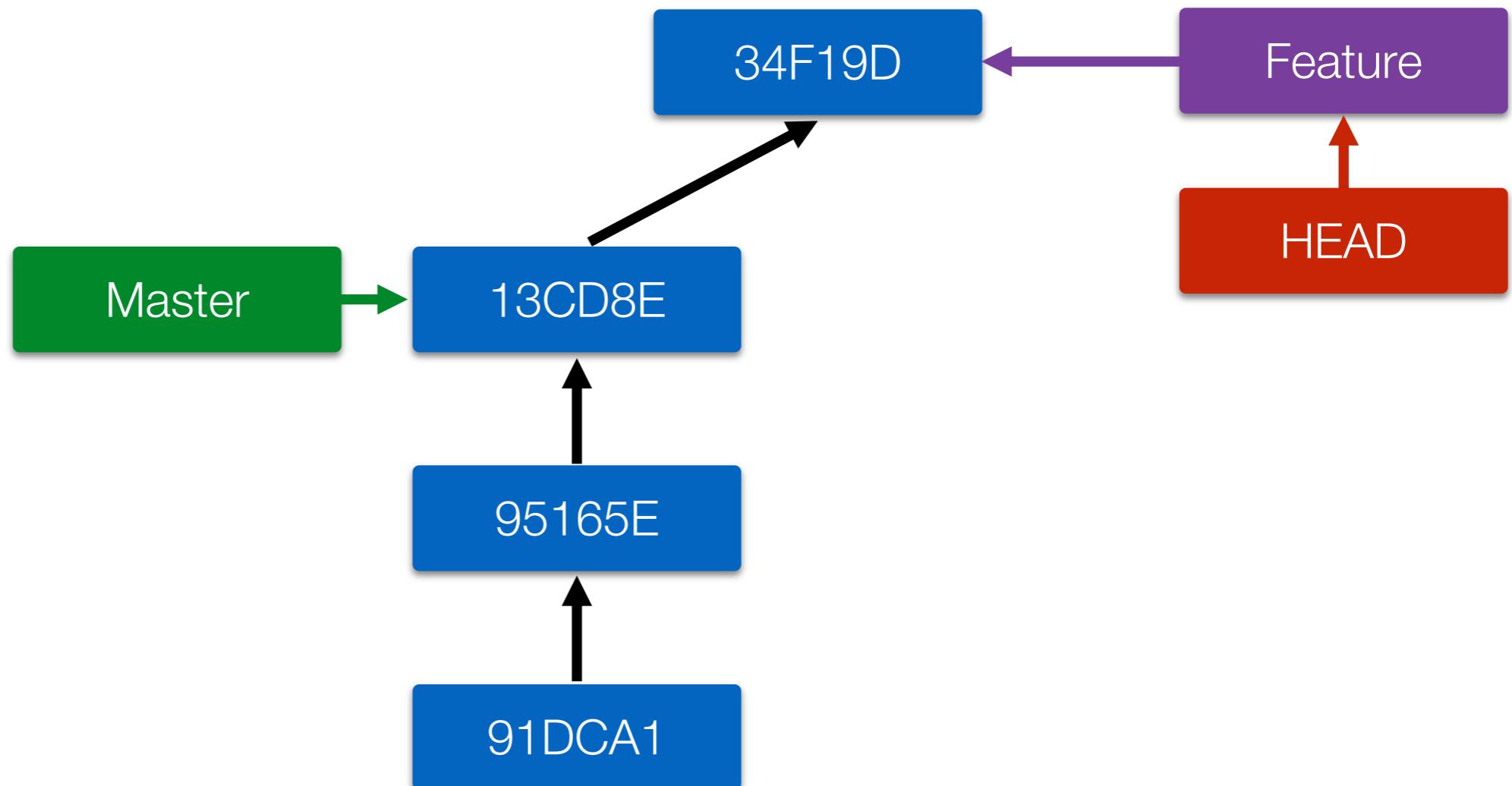
Branching



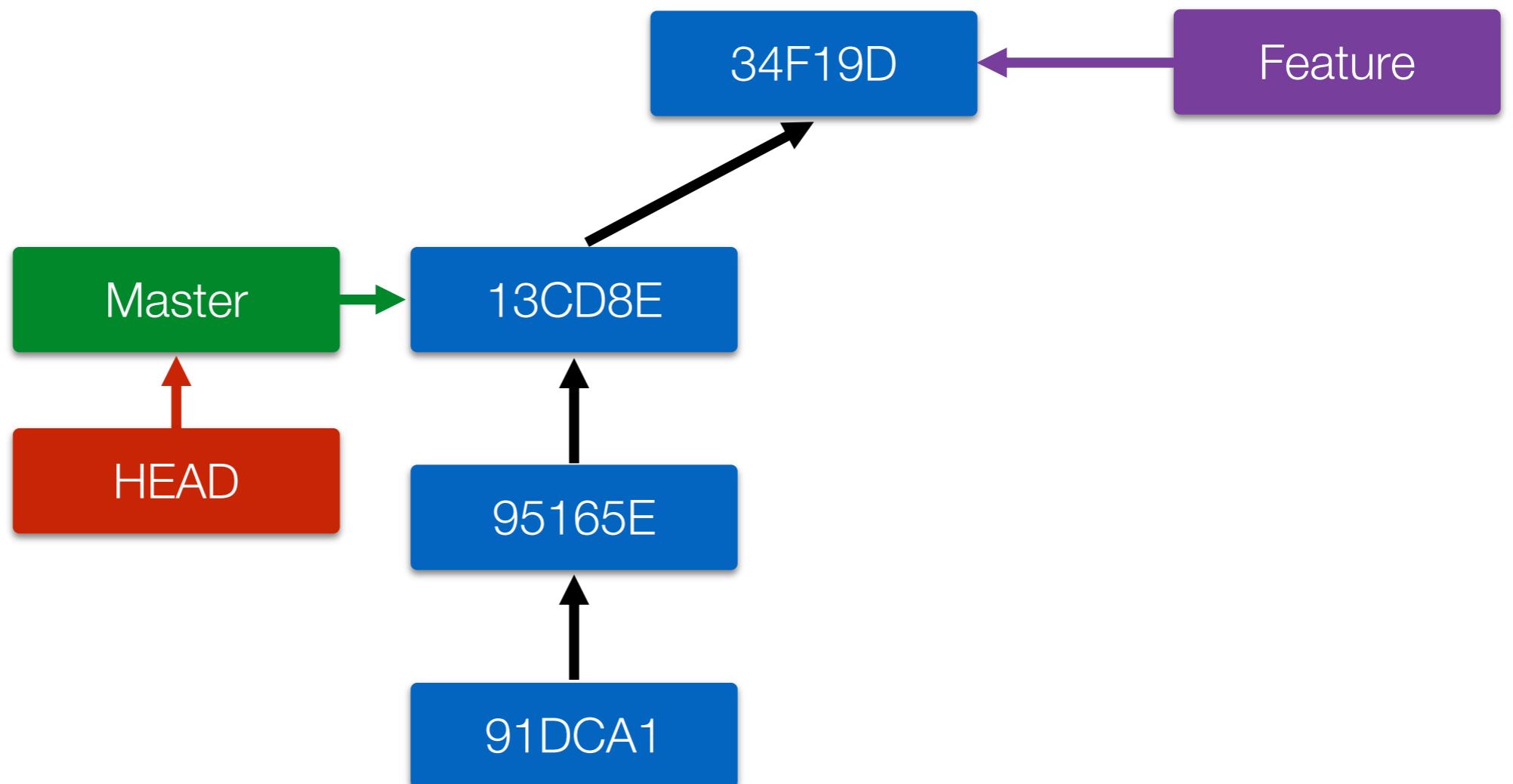
Branching



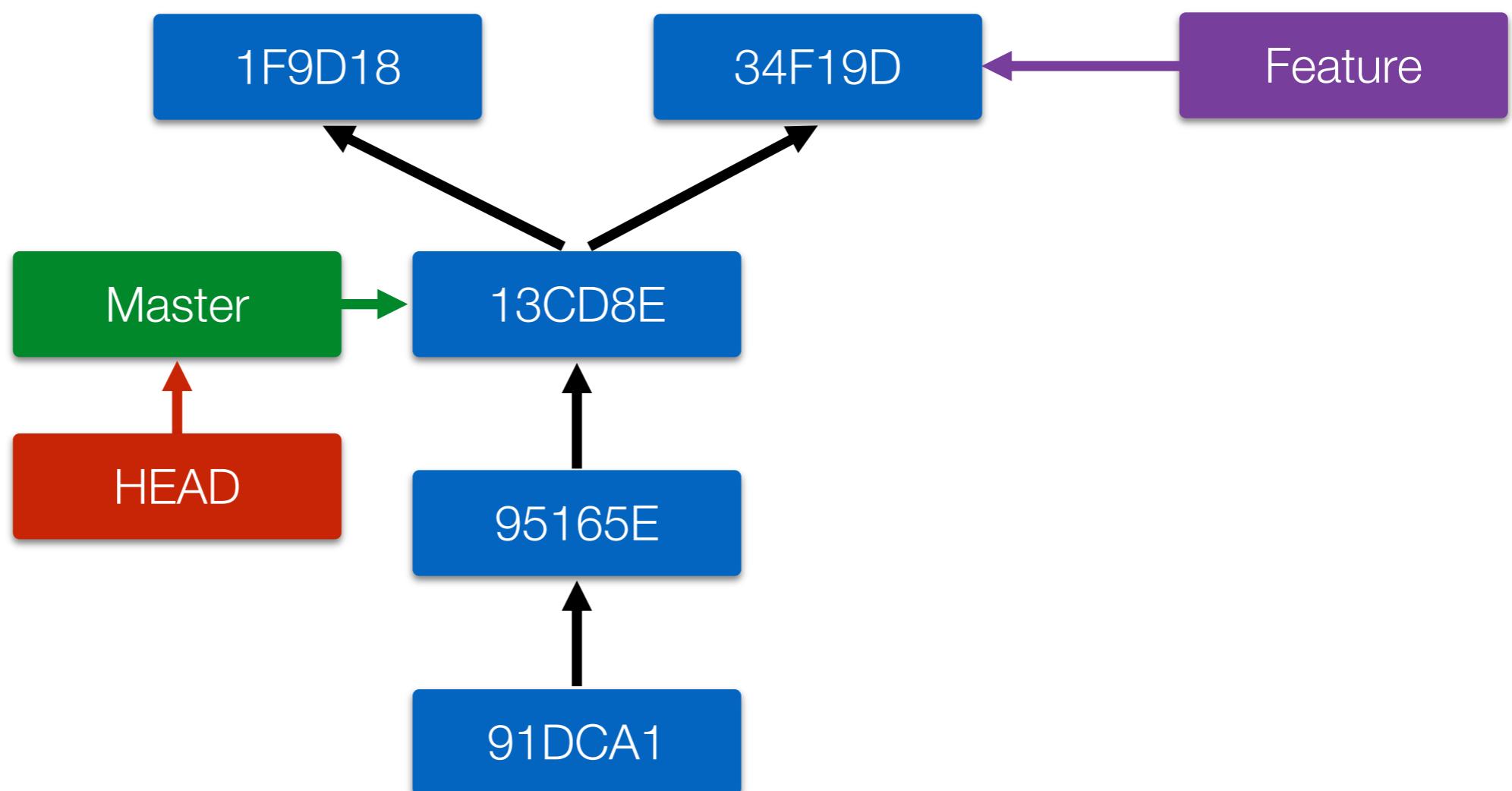
Branching



Branching



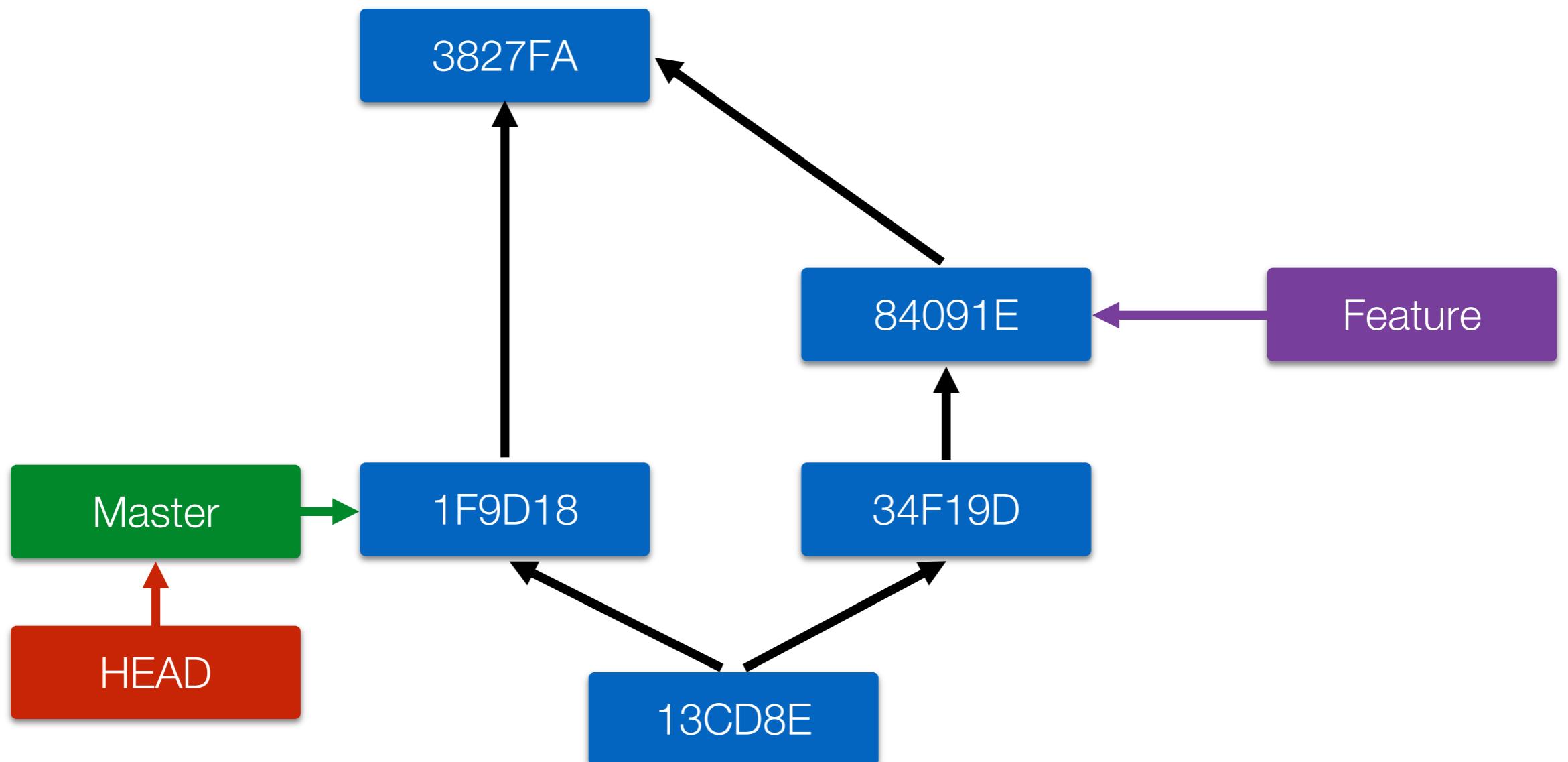
Branching



Git Branching

- Creating a new branch (label)
`git branch [branch name]`
- Checking out the branch (move the HEAD)
`git checkout [branch name]`
- Combining the above commands (create & checkout)
`git checkout -b [branch name]`

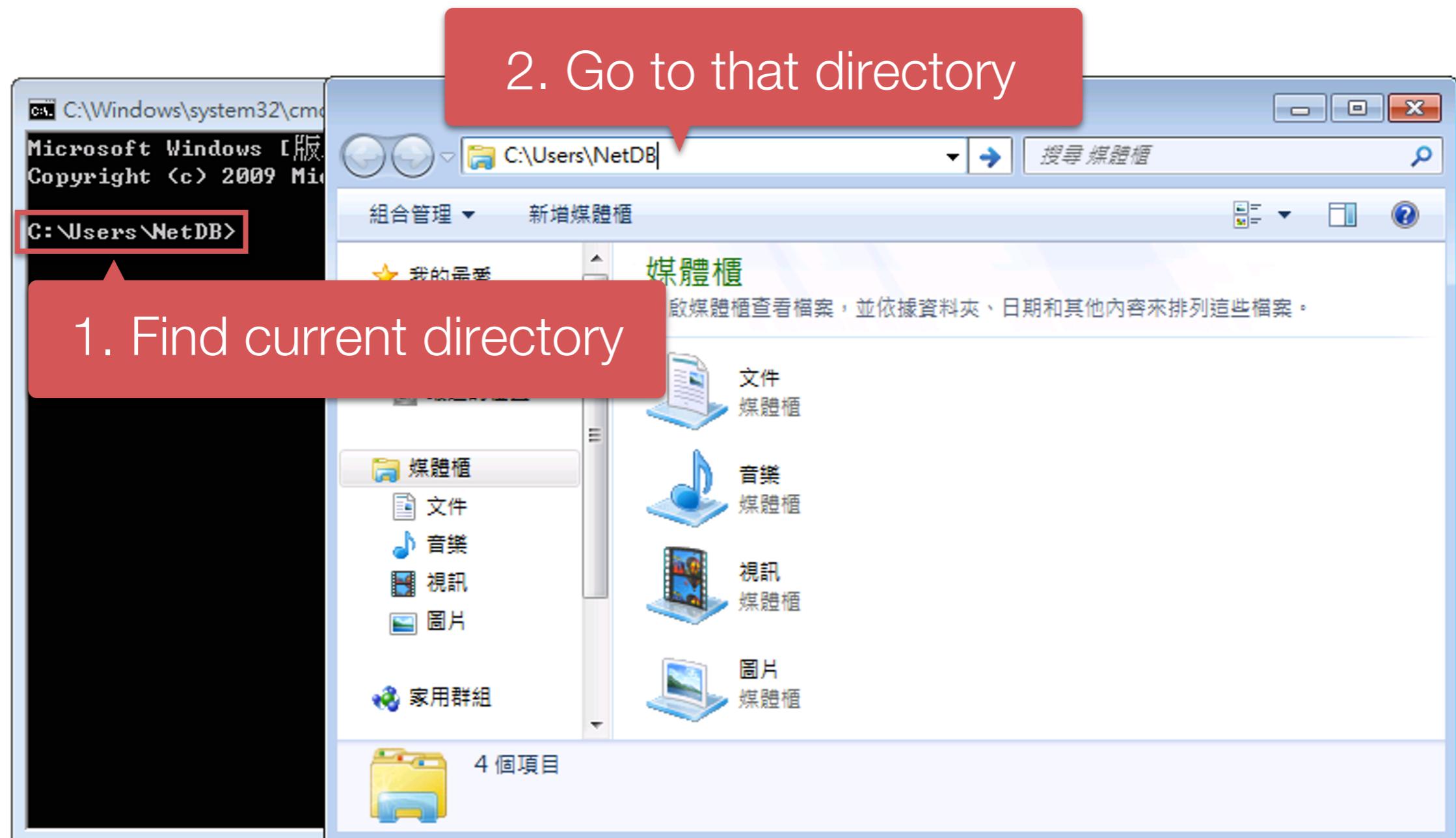
Merging

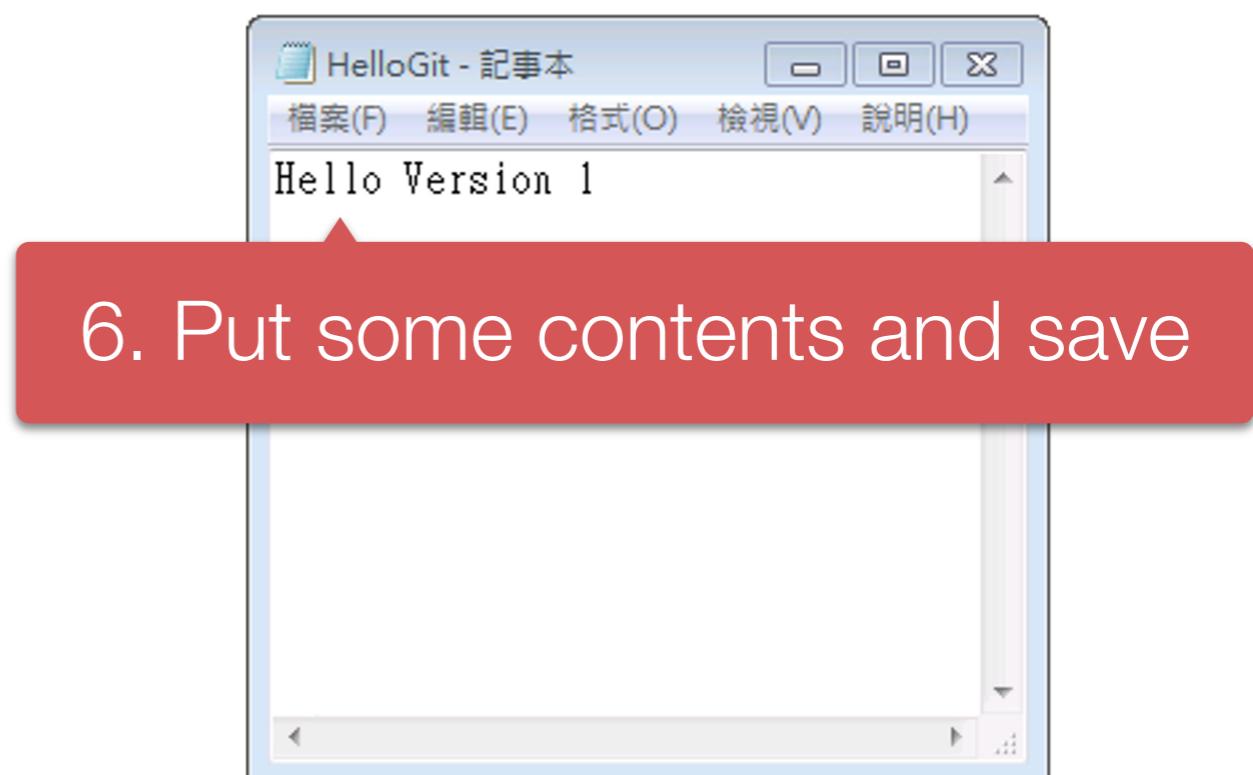
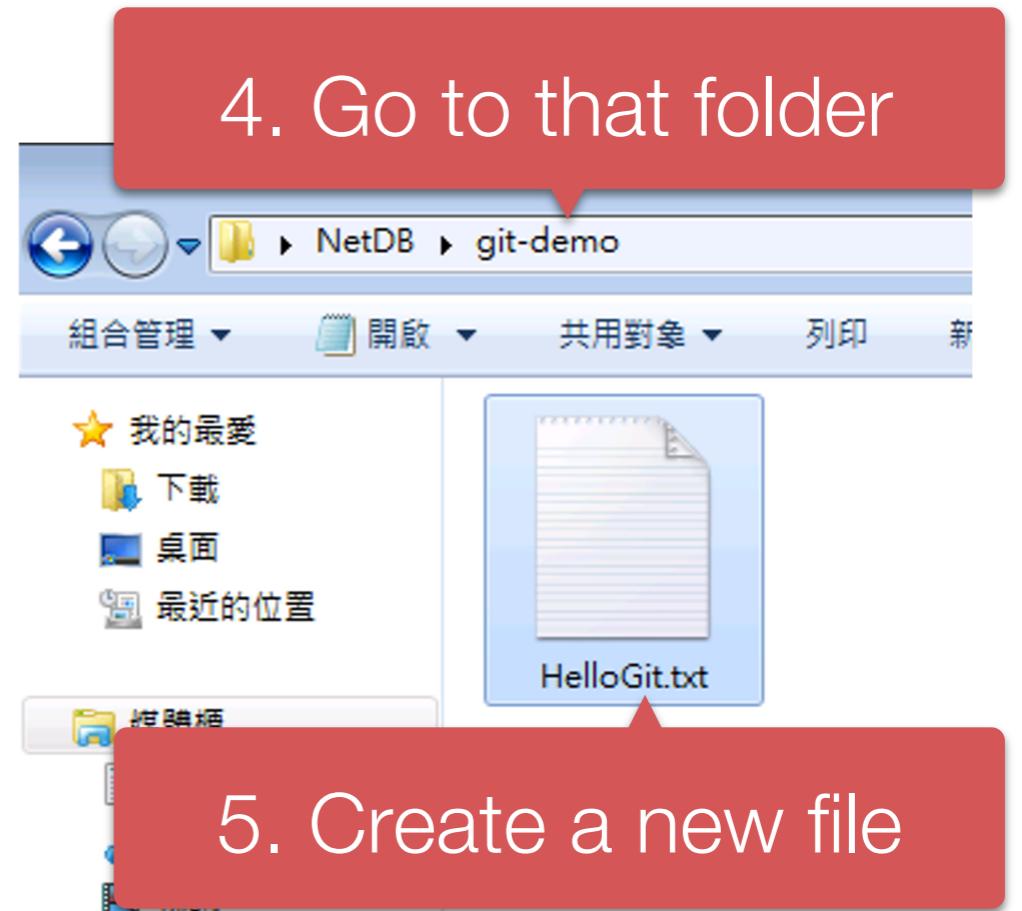
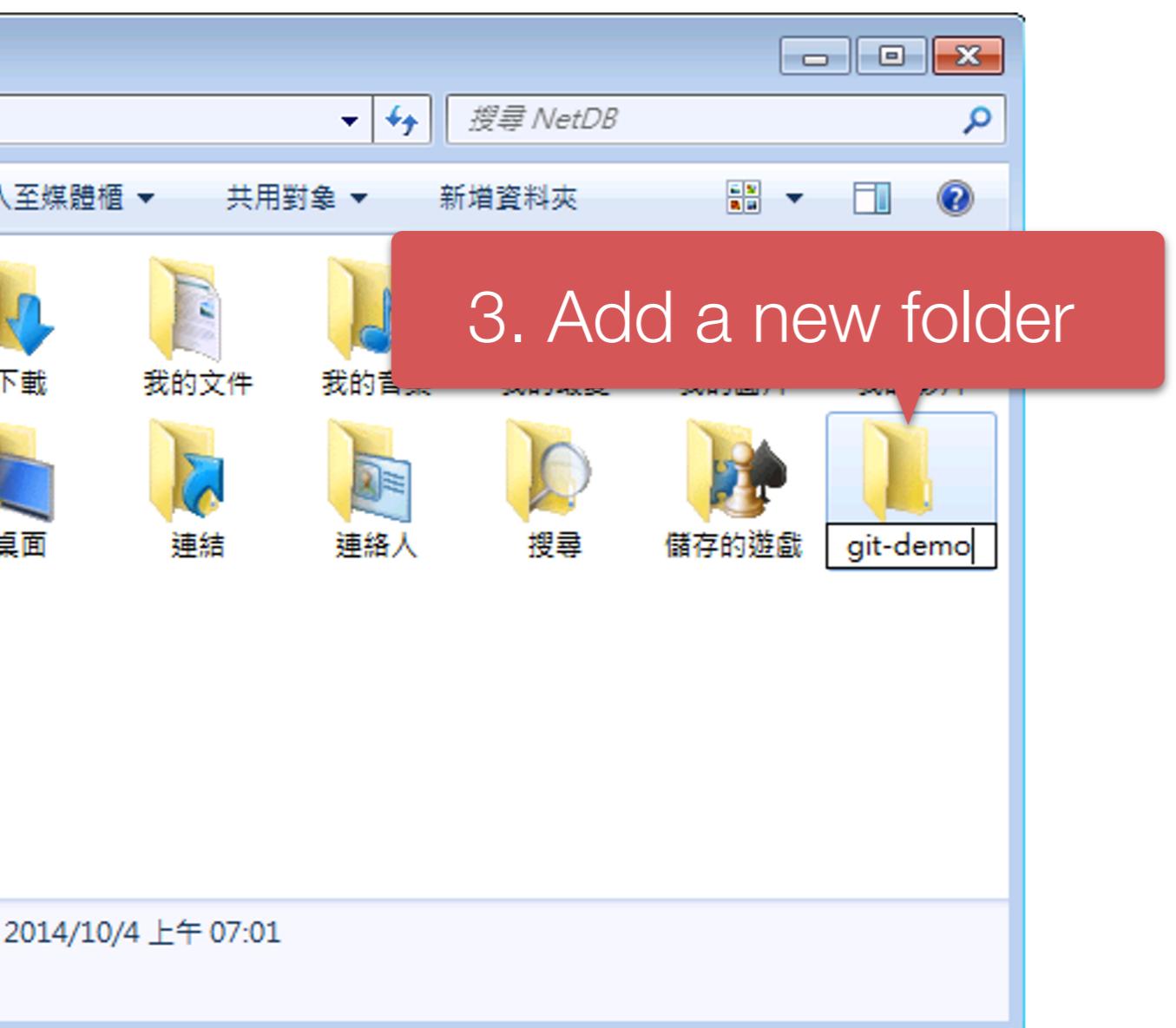


Git Merging

- Merging Steps
 - Checking out a branch to merge
`git checkout [branch 1 name]`
 - Merging another branch
`git merge [branch 2 name]`

Try Git!





```
命令提示字元  
Microsoft Windows [版本 6.1.7601]  
Copyright (c) 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:\>cd git-demo
C:\>dir

8. Go to “git-demo”

9. Show the files in “git-demo”

10. Initialize a Git repository

C:\>git init
Initialized empty Git repository in C:/Users/NetDB/git-demo/.git/

```
$ 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 init  
Initialized empty Git repository in C:/Users/NetDB/.git/  
  
C:\Users\NetDB\git-demo>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

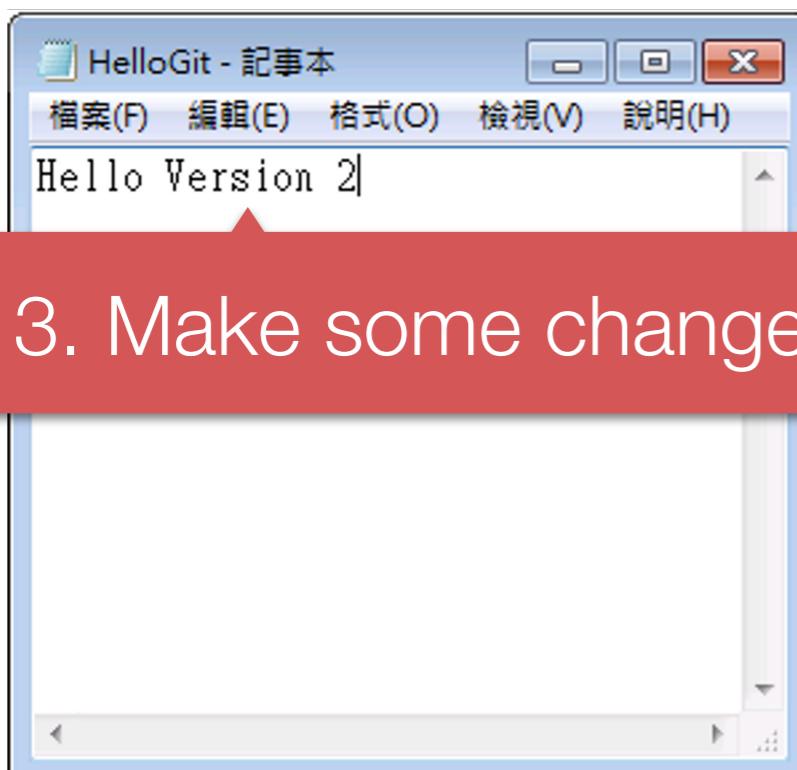
```
# Add HelloGit.txt to staging files  
$ git add HelloGit.txt
```

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

14. Add it and commit again

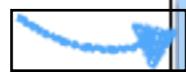
```
C:\Users\NetDB\git-demo>git add HelloGit.txt  
C:\Users\NetDB\git-demo>git commit -m "version 2"  
[master e134c84] version 2  
 1 file changed, 1 insertion(+), 1 deletion(-)  
git-demo>
```

13. Make some changes and save



15. View your versions

Version ID



```
C:\Users\NetDB\git-demo>git log  
commit e134c845df593f1451c4e9ebc874ddef6df42a76  
Author: cyhsu <cyhsu@netdb.cs.nthu.edu.tw>  
Date:   Sat Oct 4 08:09:55 2014 +0800
```

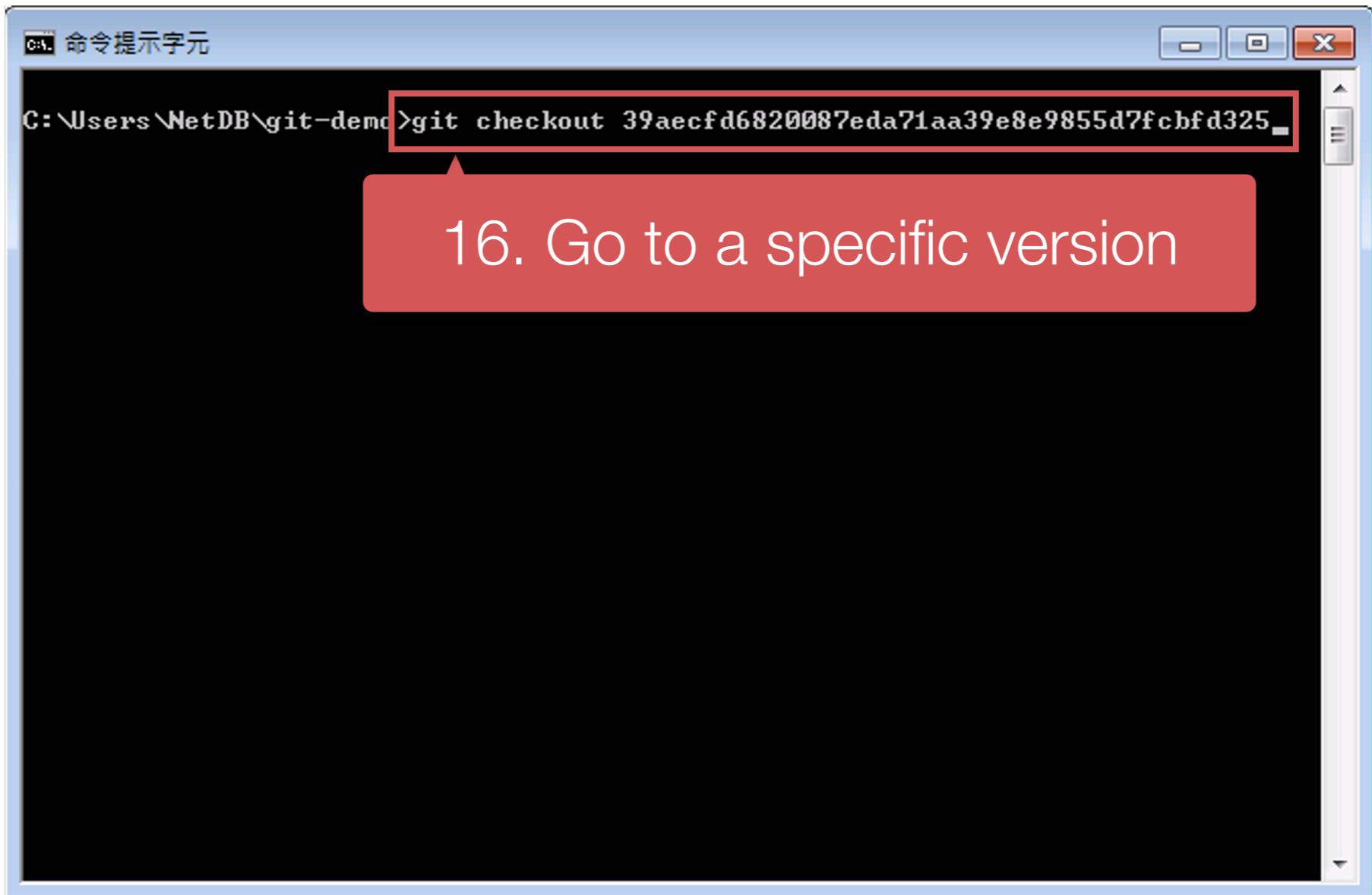
version 2

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

version 1

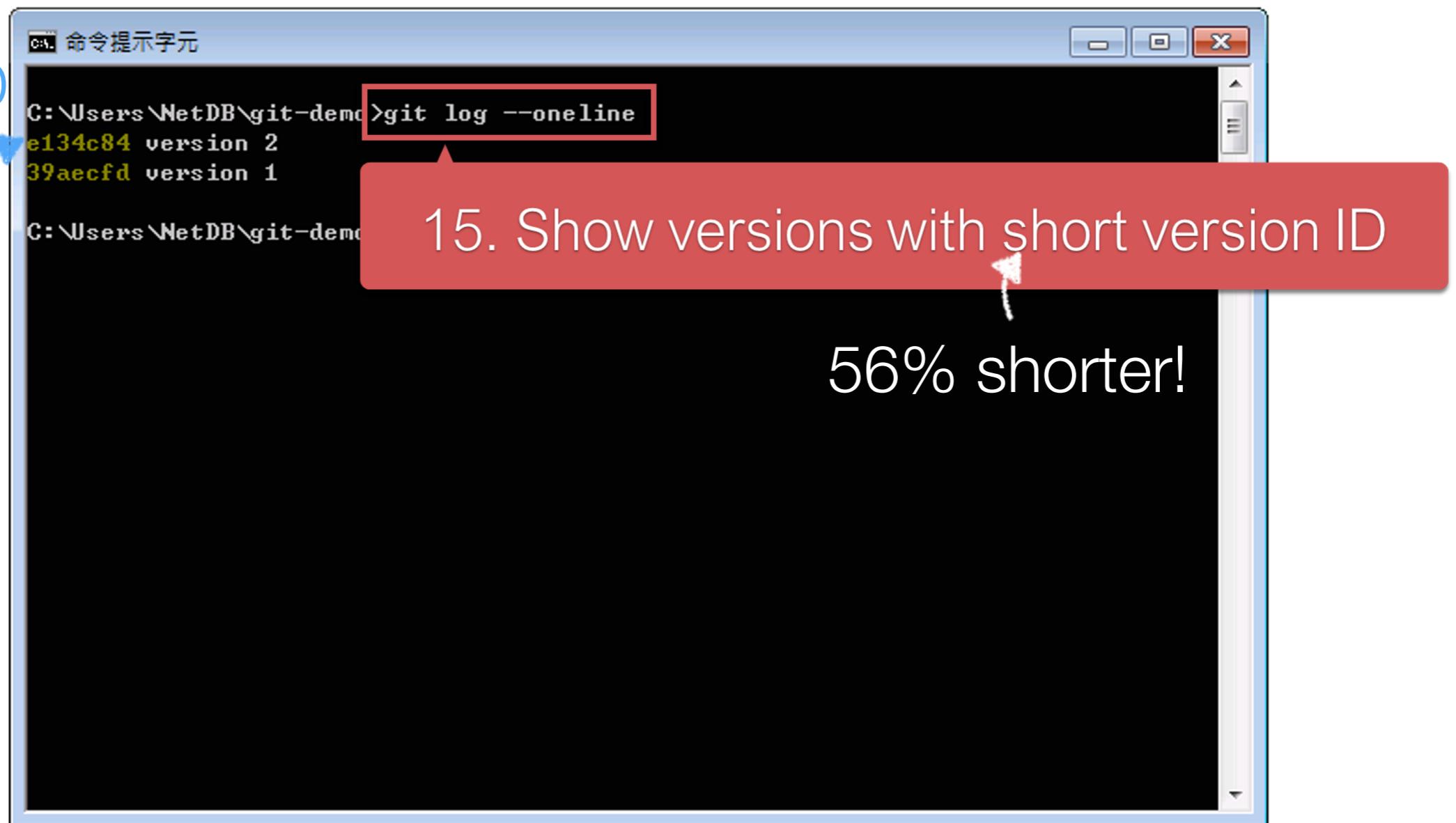
Commit messages

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



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

Version ID

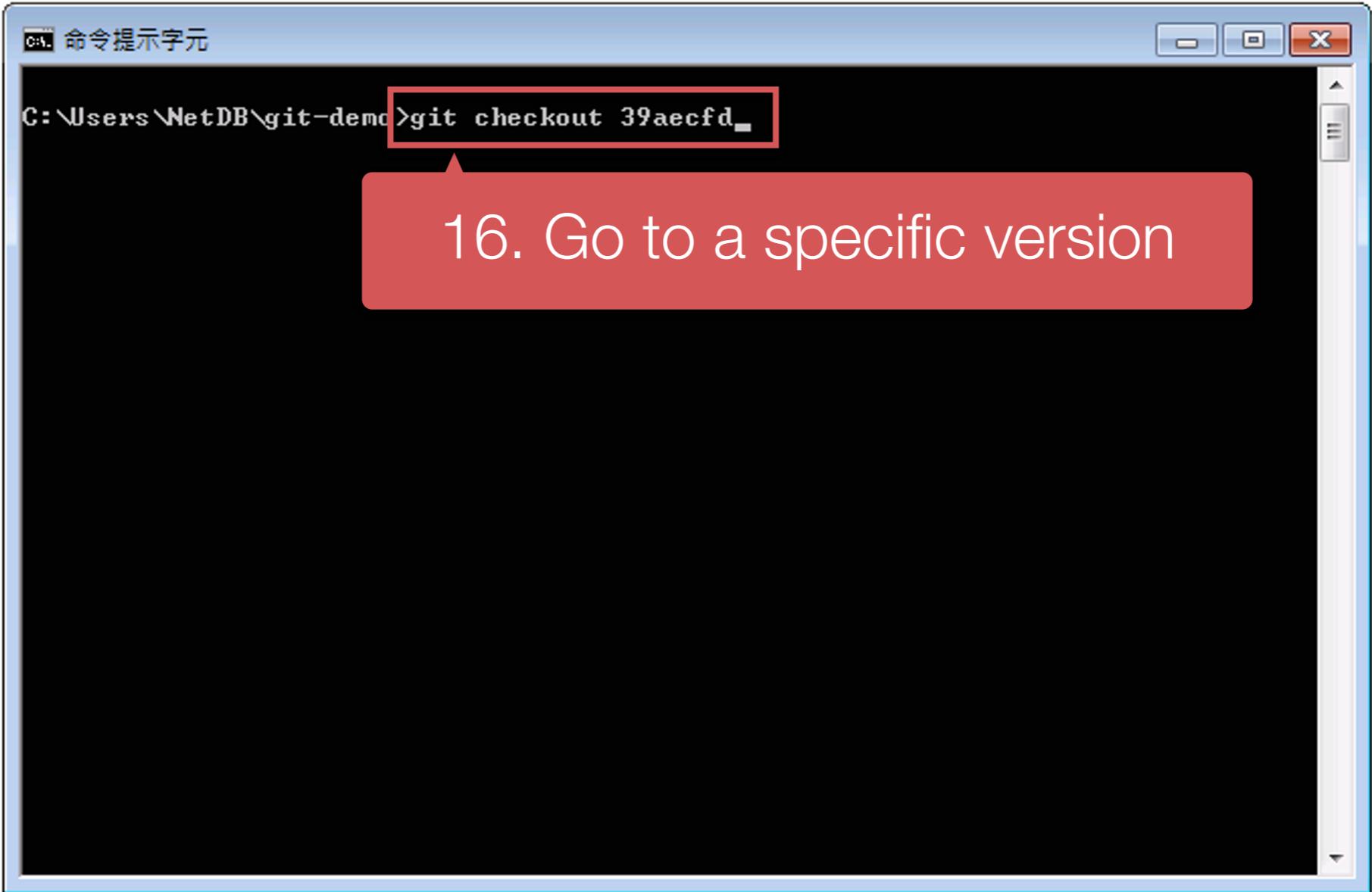


The screenshot shows a Windows Command Prompt window titled "命令提示字元" (Command Prompt). The command entered is "git log --oneline". The output displays two commits:

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

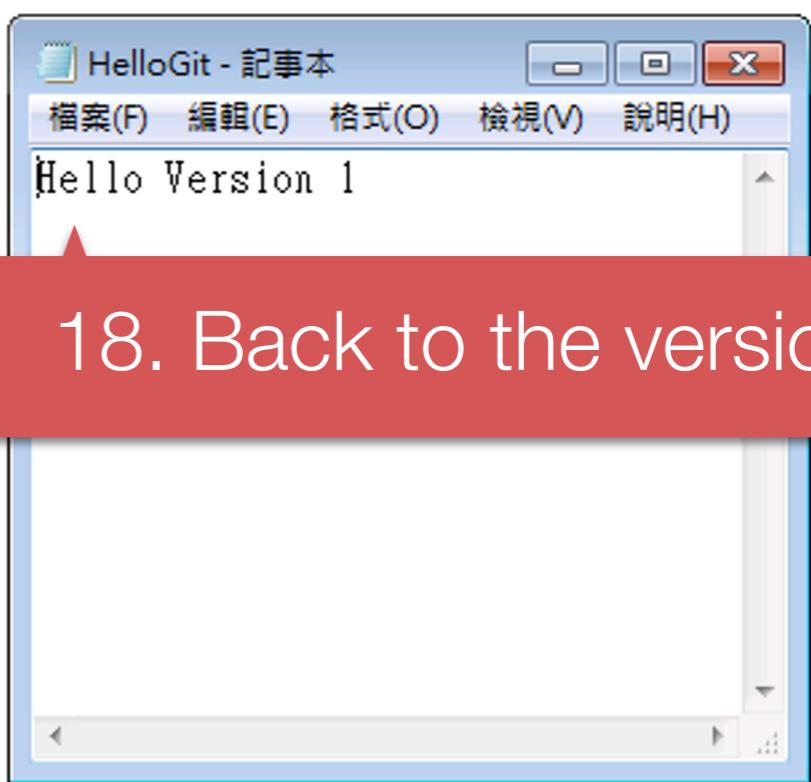
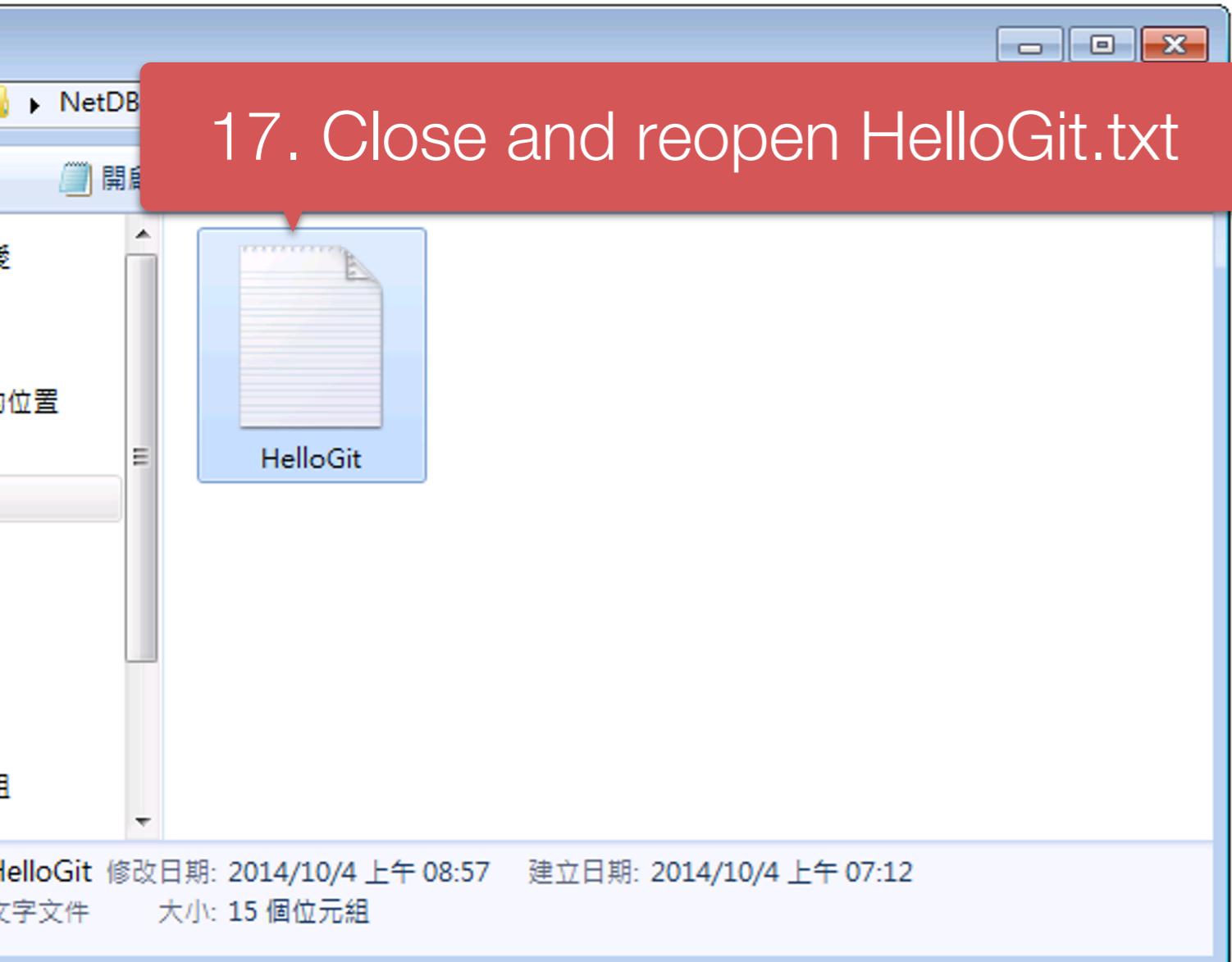
A red callout box contains the text "15. Show versions with short version ID" and "56% shorter!" with an arrow pointing to the commit IDs.

```
# 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!

Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Collaboration with Git

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

Collaboration workflow



Server

Clone



Local A

Clone



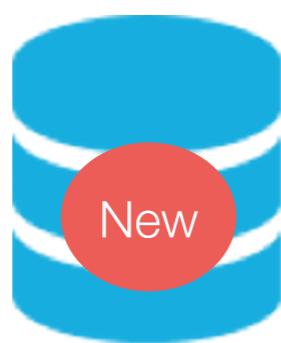
Local B

Collaboration workflow



Server

Commit



Local A



Local B

Collaboration workflow



Server

Push



Local A



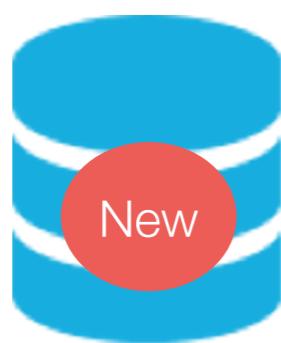
Local B

Collaboration workflow



Server

Push



Local A

Pull



Local B

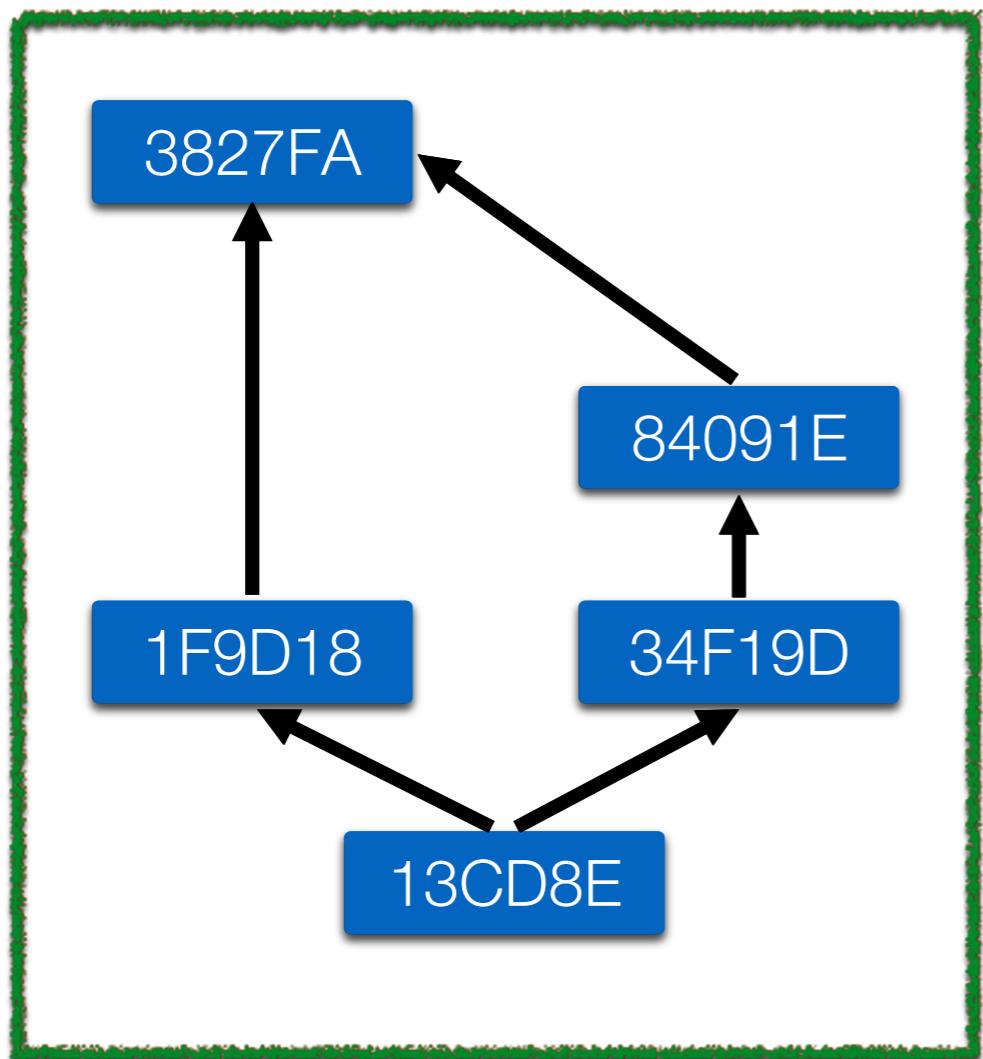
Cloning & Pushing

- Cloning the remote repositories
`git clone [Remote URL]`
- The [Remote URL] is saved as **Origin**
 - After committing a few versions, you can push the branch back to **Origin**
`git push -u origin [Branch Name]`

Fetch & Pull

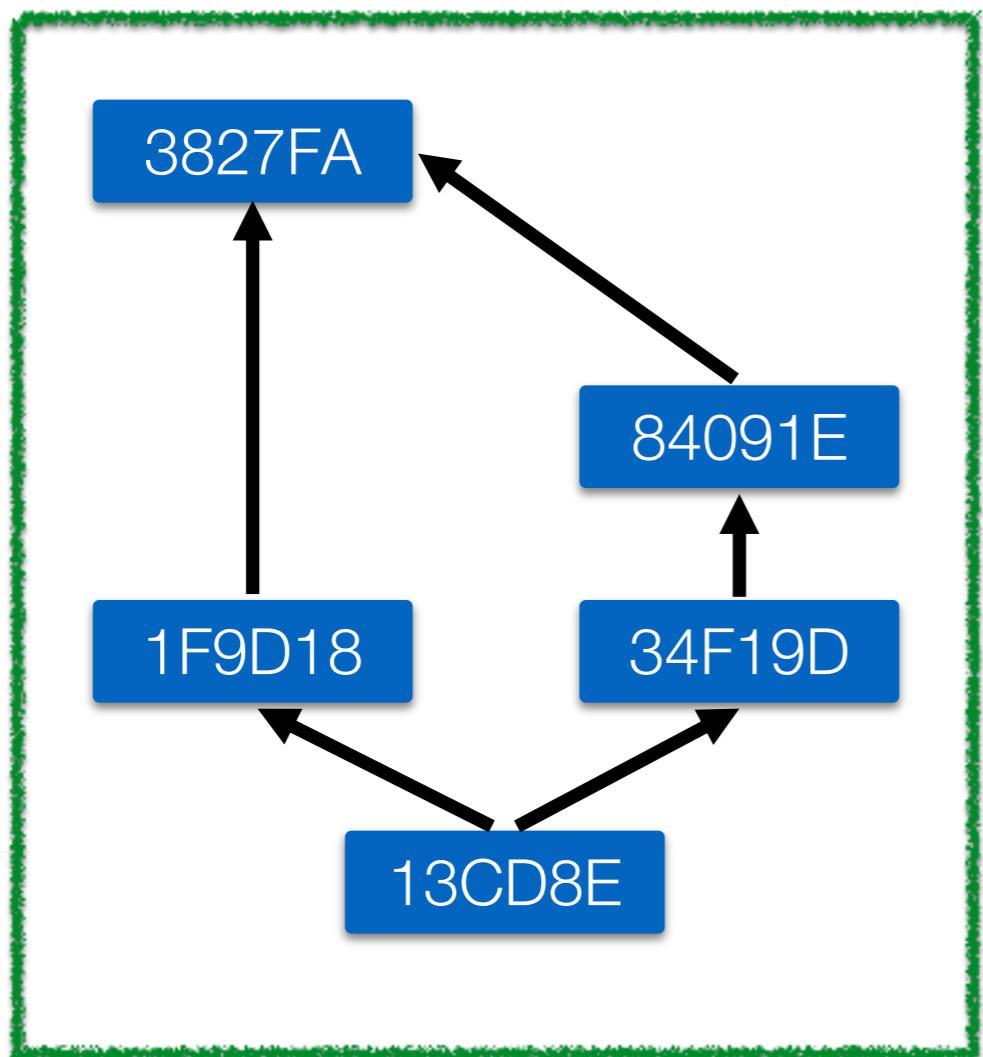
- Updating a branch from the remote repository
 - Fetching the remote repository to local
`git fetch origin`
 - Merging the remote branch
`git merge origin/[Branch Name]`
- Doing above commands in one command
`git pull [Branch Name]`

Fork

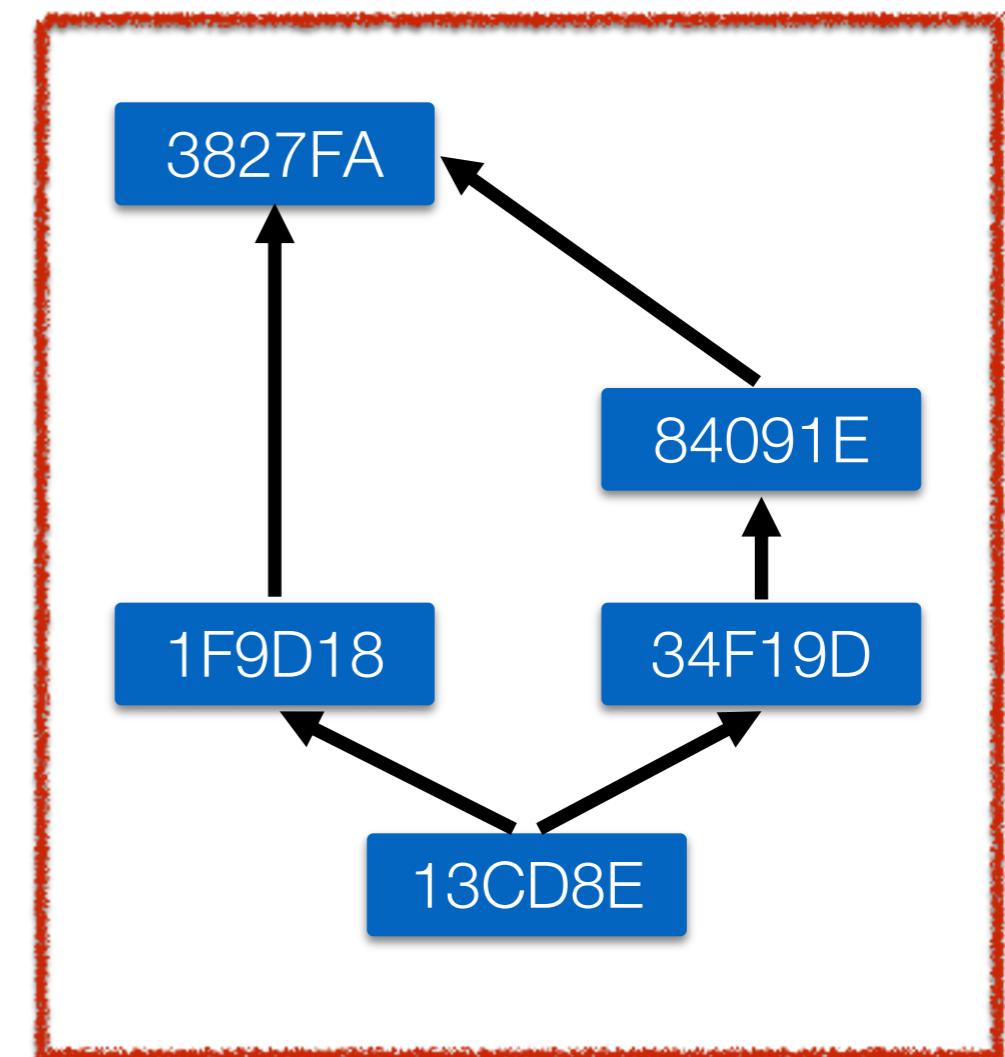


The Repo. Under TA's Account

Fork

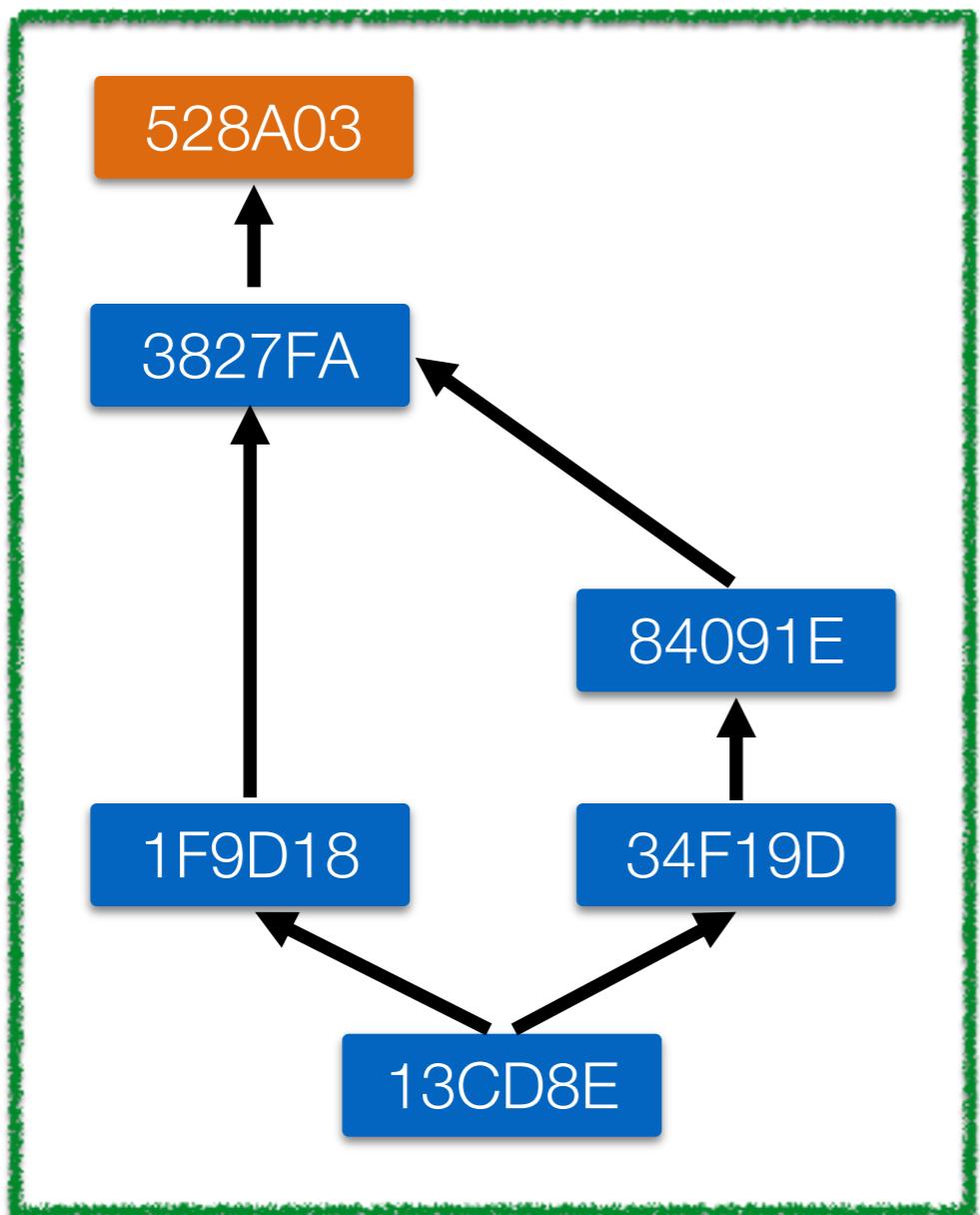


The Repo. Under TA's Account



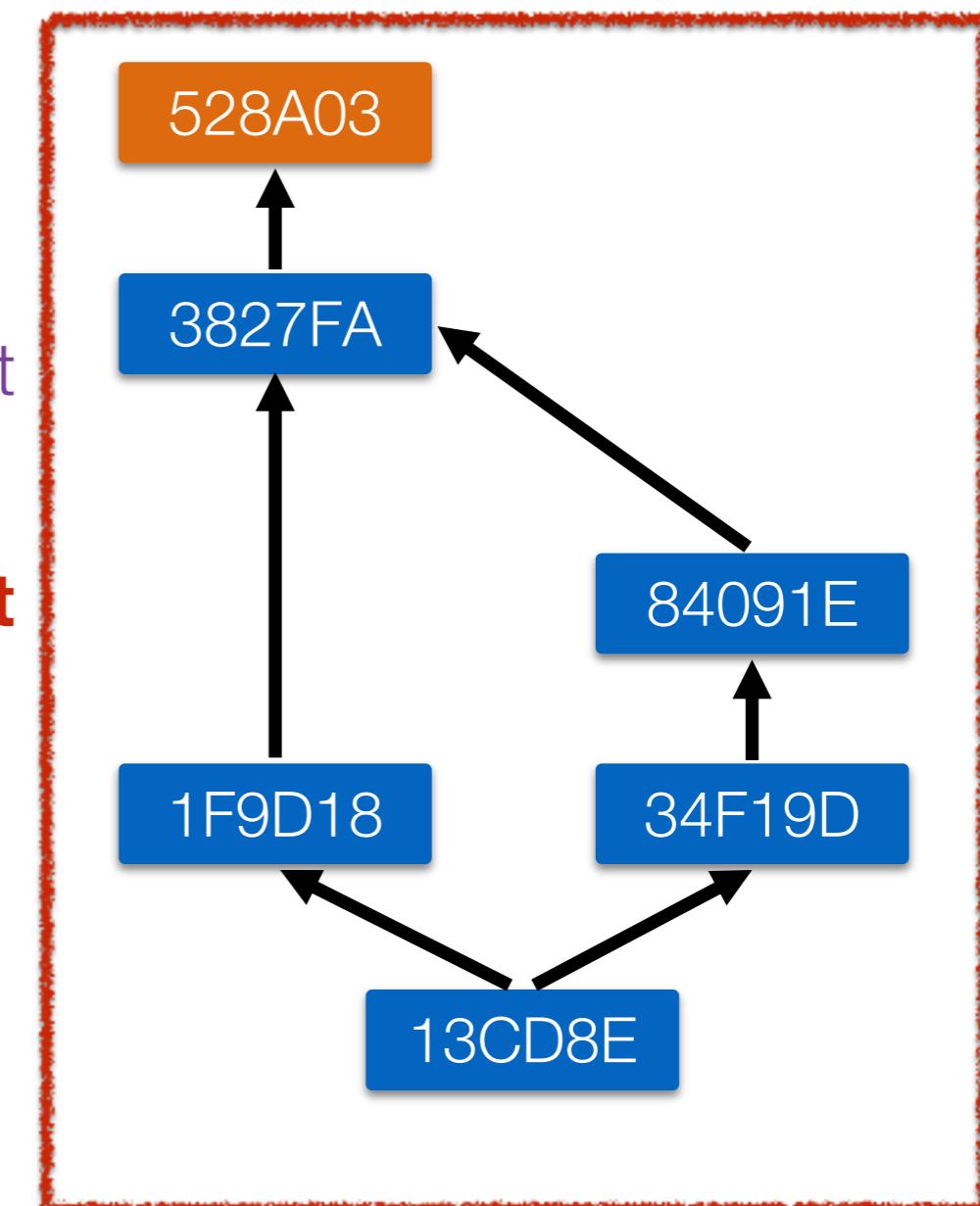
The Repo. Under Your Account

Pull (Merge) Request



The Repo. Under TA's Account

Pull
Request
Accept



The Repo. Under Your Account

.gitignore File

- You can ignore some files that you don't want them to be tracked by editing the .gitignore file
- Remember to track and commit your .gitignore file
- Don't know what should be in .gitignore ?
 - <https://github.com/github/gitignore>
 - <https://www.gitignore.io/>

How to Submit Your Code to 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

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 Projects Groups Activity Milestones Snippets

Search or jump to...

P practice-submission

courses > 2020-spring > practice-submission > Details

practice-submission Project ID: 9897

No license. All rights reserved 1 Commit 1 Branch 0 Tags 41 KB Files

master practice-submission / + History Find file Web IDE

Initial commit
Yi-Chun Chen authored 6 hours ago 9af13851

README Auto DevOps enabled

Name	Last commit	Last update
README.md	Initial commit	6 hours ago

README.md

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 an assignment or a project.

Try It !!

Collapse sidebar

GitLab Projects Groups Activity Milestones Snippets

Search or jump to...

P practice-submission

Pin-Yu Wang > practice-submission > Details

practice-submission Project ID: 9902

Add license 0 Commits 1 Branch 0 Tags 0 Bytes Files

Forked from courses / databases / 2020-spring / practice-submission

master practice-submission + History Find file Web IDE

Initial commit
Yi-Chun Chen authored 6 hours ago 9af13851

README Add CHANGELOG Add CONTRIBUTING Auto DevOps enabled

Name	Last commit	Last update
README.md	Initial commit	6 hours ago

README.md

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 an assignment or a project.

1. Click to fork



GitLab Projects Groups Activity Milestones Snippets

Search or jump to...

P practice-submission

Project ID: 3362

2. Check if this repository is under your account

Add license 0 Commits 1 Branch 0 Tags 0 Bytes Files

Forked from courses / databases / 2020-spring / practice-submission

master practice-submission + History Find file Web IDE

Initial commit
Yi-Chun Chen authored 6 hours ago 9af13851

README Add CHANGELOG Add CONTRIBUTING Auto DevOps enabled

Name	Last commit	Last update
README.md	Initial commit	6 hours ago

README.md

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 an assignment or a project.

3. Go to settings

GitLab Projects Groups Activity Milestones Snippets

Pin-Yu Wang > practice-submission > General Settings

General project

Update your project name, description, avatar, and other general settings.

Permissions

Enable or disable certain permissions for this project.

Project visibility

Private

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

Issues

Lightweight issue tracking system for this project

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

4. Set project to private

P practice-submission

Project Repository Issues Merge Requests Wiki Snippets

Settings

General

Members Integrations Repository

Collapse sidebar

This screenshot shows the 'General Settings' page for a project named 'practice-submission' on GitLab. The left sidebar is visible with various project management options like Project, Repository, and Settings. The main content area shows the 'General project' section with a brief description and the 'Permissions' section. A prominent red callout bubble with the text '4. Set project to private' points to the 'Project visibility' dropdown menu, which is currently set to 'Private'. Below this, there are sections for 'Issues', 'Repository', and 'Merge requests', each with its own permission settings. The 'Repository' section is expanded, showing that only project members can view and edit files. The 'Merge requests' section also has a setting where only project members can submit changes upstream.

GitLab Projects Groups Activity Milestones Snippets

Search or jump to...

P practice-submission

Project Repository Issues (0) Merge Requests (0) Wiki Snippets

Settings

General Members Integrations Repository

5. Scroll down and save changes

Save changes

Merge request
Customize your merge request restrictions.

Badges
Customize your project badges. [Learn more about badges.](#)

Export project
Export this project with all its related data in order to move your project to a new GitLab instance. Once the export is finished, you can import the file from the "New Project" page.

Advanced
Perform advanced options such as housekeeping, archiving, renaming, transferring, or removing your project.

<< Collapse sidebar

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

The screenshot shows a GitLab project page for 'practice-submission'. The left sidebar contains navigation links: Project, Details, Activity, Releases, Cycle Analytics, Repository, Issues (0), Merge Requests (0), Wiki, Snippets, and Settings. The main content area displays the project details, including the title 'practice-submission' (Project ID: 9902), a summary of 0 Commits, 1 Branch, 0 Tags, and 0 Bytes Files, and a note that it was Forked from 'courses / databases / 2020-spring / practice-submission'. Below this is a commit history showing an 'Initial commit' by Yi-Chun Chen. At the bottom, there are options to add README, CHANGELOG, CONTRIBUTING, and Auto DevOps, and a file list showing 'README.md' with its details: Name, Last commit (Initial commit), and Last update (6 hours ago). A red callout box labeled '1. Choose HTTPS' points to the HTTPS cloning link, and another red callout box labeled '2. Copy the link' points to the SSH cloning link.

Pin-Yu Wang > practice-submission > Details

practice-submission

Project ID: 9902

Add license 0 Commits 1 Branch 0 Tags 0 Bytes Files

Forked from courses / databases / 2020-spring / practice-submission

master practice-submission / +

Initial commit
Yi-Chun Chen authored 6 hours ago

9af13851

1. Choose HTTPS

Clone with SSH
git@shwu10.cs.nthu.edu.tw:pywang/practice-submission

Clone with HTTPS
https://shwu10.cs.nthu.edu.tw/p/9af13851

2. Copy the link

README Add CHANGELOG Add CONTRIBUTING Auto DevOps enabled

Name	Last commit	Last update
README.md	Initial commit	6 hours ago

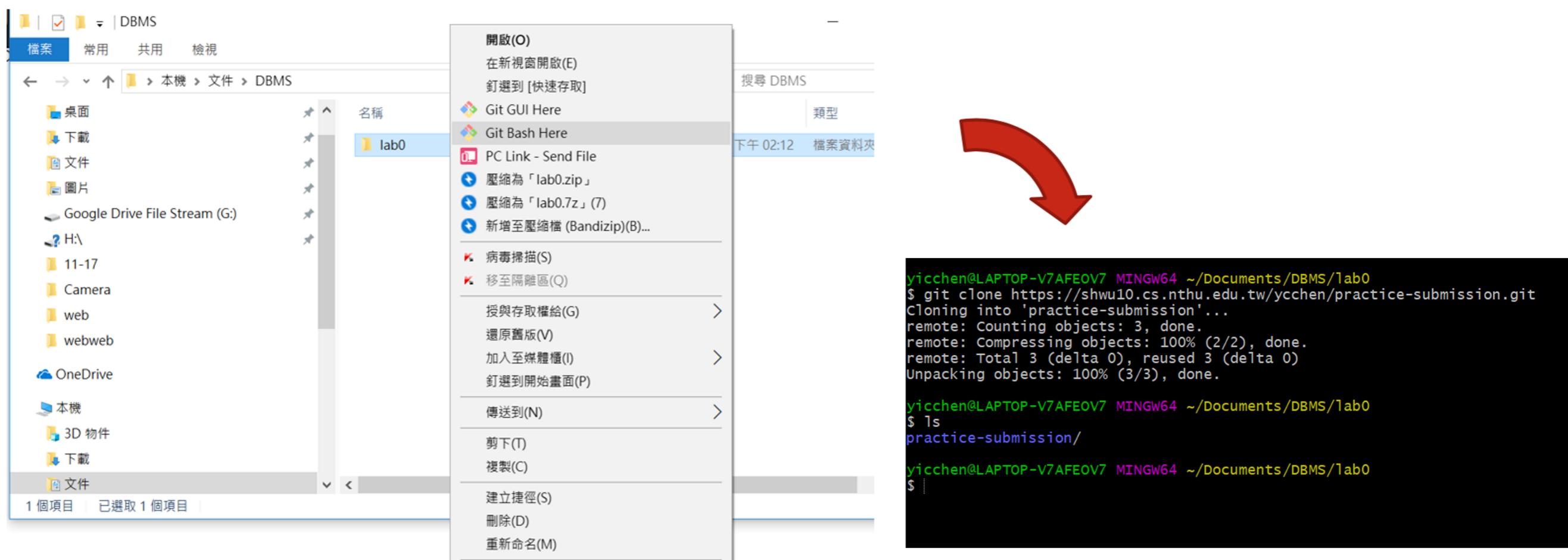
README.md

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 process.

Collaborator sidebar
https://shwu10.cs.nthu.edu.tw/pywang/practice-submission#

If You use Windows



3. Create a folder to put your repos

```
yicchen@LAPTOP-V7AEOFV7 MINGW64 ~/Documents/DBMS/1ab0
$ git clone https://shwu10.cs.nthu.edu.tw/ycchen/practice-submission.git
Cloning into 'practice-submission'...
remote: Counting objects: 3, done.    4. Type "git clone {URL}"
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0)
Unpacking objects: 100% (3/3), done.
```

```
yicchen@LAPTOP-V7AEOFV7 MINGW64 ~/Documents/DBMS/1ab0
$ ls
practice-submission/
```

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

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

```
yicchen@LAPTOP-V7AEOFV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git add -A
```

1. -A means all files

```
yicchen@LAPTOP-V7AEOFV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git status
```

on branch master

Your branch is up to date with 'origin/master'.

2. Check if your file is added to git

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: practice.txt

```
yicchen@LAPTOP-V7AEOFV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git commit -m "Finish"
```

[master 93a03d5] Finish

1 file changed

create mode 100644

3. Commit your changes

```
yicchen@LAPTOP-V7AEOFV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git commit -m "Finish"
```

```
*** Please tell me who you are.
```

```
Run
```

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

```
to set your account's default identity.
Omit --global to set the identity only in this repository
```

```
fatal: unable to auto-detect email address (got 'yicchen@LAPTOP-V7AEOFV7.(none)')
```

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

```
yicchen@LAPTOP-V7AFOV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git push origin master
Counting objects: 3, done.
Delta compression
Compressing object
Writing objects: 100% (3/3), 284 bytes | 284.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://shwu10.cs.nthu.edu.tw/yccchen/practice-submission.git
 21e2fda..93a03d5  master -> 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

The screenshot shows a GitLab project page for 'practice-submission'. The left sidebar contains navigation links: Project, Details, Activity, Releases, Cycle Analytics, Repository, Issues (0), Merge Requests, Wiki, Snippets, and Members. A red callout box highlights the 'Merge Requests' link with the text '1. Click Merge Requests'. The main content area displays project details: No license, 1 Commit, 1 Branch, 0 Tags, and 41 KB Files. It shows a commit titled 'Initial commit' by 'Yi-Chun Chen' from 6 hours ago, with a commit hash '9af13851'. Below the commit is a file list for 'README.md' showing its last commit was an 'Initial commit' 6 hours ago. The page also includes a 'Practice Submission' section with instructions for practicing submissions.

courses > 2020-spring > practice-submission > Details

practice-submission Project ID: 9897

No license. All rights reserved 1 Commit 1 Branch 0 Tags 41 KB Files

master practice-submission / + History Find file Web IDE

Initial commit
Yi-Chun Chen authored 6 hours ago
9af13851

Name Last commit Last update

README.md Initial commit 6 hours ago

README.md

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 an assignment or a project.

GitLab Projects Groups Activity Milestones Snippets

Search or jump to...       

P practice-submission

Project Repository Issues Merge Requests Wiki Snippets Members

courses > ... > 2020-spring > practice-submission > Merge Requests



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

2. New merge request

The screenshot shows a GitLab interface for a project named 'practice-submission'. On the left, there's a sidebar with options like Project, Repository, Issues, Merge Requests, Wiki, Snippets, and Settings. The main area is titled 'New Merge Request'.

Step 3: A red callout points to the 'Source branch' dropdown, which is set to 'pywang/practice-submission' and 'master'. Below it, a commit history shows a 'first commit' by Pin-Yu Wang.

Step 4: Another red callout points to the 'Target branch' dropdown, which is set to 'courses/databases/2020-spring/practic...' and 'master'. Below it, a commit history shows an 'Initial commit' by Yi-Chun Chen.

Step 5: A red callout at the bottom points to a green button labeled 'Compare branches and continue'.

GitLab Projects Groups Activity Milestones Snippets

+ This project Search

P practice-submission

Project Repository Issues Merge Requests Wiki Snippets Settings

New Merge Request

From ycchen/practice-su...

6. Set title to "{ID} Submission"

Title std107000000 Submission

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

Add [description templates](#) to help your contributors communicate effectively!

Description

Write Preview

finish|

Markdown and quick actions are supported

Attach a file

Assignee Assignee [Assign to me](#)

Milestone Milestone

Labels Labels

Source branch master

Target branch 107000000 [Change branches](#)

Squash commits when merge request is accepted. [About this feature](#)

A screenshot of the GitLab 'New Merge Request' interface. On the left, there's a sidebar with project navigation links like Project, Repository, Issues, Merge Requests, Wiki, Snippets, and Settings. The main area shows a 'New Merge Request' form. At the top, it says 'From ycchen/practice-su...'. A large red callout bubble points to the 'Title' field, which contains 'std107000000 Submission'. Below the title is a note about starting with 'WIP:' and a link to 'description templates'. The 'Description' section has a rich-text editor with a 'Write' tab active, showing the word 'finish'. There are buttons for 'Preview', bold, italic, and other rich-text controls. Below the editor is a note about Markdown support and a 'Attach a file' button. Further down, there are fields for 'Assignee' (set to 'Assignee'), 'Milestone' (empty), 'Labels' (empty), 'Source branch' (set to 'master'), and 'Target branch' (set to '107000000'). A checkbox for 'Squash commits when merge request is accepted' is at the bottom. The top right of the interface includes standard GitLab navigation icons like '+' and search.

GitLab Projects Groups Activity Milestones Snippets + This project Search

P practice-submission

Project Repository Issues 0 Merge Requests 0 Wiki Snippets Settings

Assignee Assignee Assign to me

Milestone Milestone

Labels Labels

Source branch master

Target branch 107000000 Change branches

Squash commits when merge request is accepted. [About this feature](#)

Contribution Allow commits from members who can merge to the target branch. [About this feature](#)

Submit merge request

Commits 1 Changes 1

24 Feb, 2019 1 commit

Finish Your Name authored 55 minutes ago

93a03d56

Markdown and quick actions are supported Attach a file

7. If everything is OK, submit your merge request

Cancel

« Collapse sidebar

Attach a file

P practice-submission

Project overview

Repository

Issues 0

Merge Requests 0

Operations

Packages & Registries

Analytics

Wiki

Snippets

Members

Settings

chanchishen > practice-submission > Members

Project members

You can invite a new member to **practice-submission** or invite another group.

Invite member **Invite group**

GitLab member or Email address
Yu-Xuan Lin x

Choose a role permission
Maintainer v

Read more about role permissions

Access expiration date
Expiration date

1. Click Members

Members 1

GitLab Projects Groups More Search or jump to... ?

P practice-submission chanchishen > practice-submission > Members

Project members You can invite a new member to **practice-submission** or invite another group.

Invite member Invite group

GitLab member or Email address: Yu-Xuan Lin

Choose a role permission: Maintainer

Access expiration date: Expiration date

Members 1

2. Set your members

3. Choose your members to be Maintainer

4. Click Invite

The screenshot shows the 'Members' section of a GitLab project named 'practice-submission'. On the left sidebar, the 'Members' option is selected. The main area displays the 'Project members' section with a message about inviting new members. A red callout box labeled '2. Set your members' points to the 'GitLab member or Email address' input field where 'Yu-Xuan Lin' is entered. Another red callout box labeled '3. Choose your members to be Maintainer' points to the 'Choose a role permission' dropdown set to 'Maintainer'. A third red callout box labeled '4. Click Invite' points to the green 'Invite' button at the bottom of the form. The status bar at the bottom shows 'Members 1'.

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>