

GitLab for students

GitLab login and initial setup

1. At hconnect.hanyang.ac.kr, click "Sign in with Hanyang"



You need to sign in or sign up before continuing.

GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in

Username or email

Password

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

Sign in

Sign in with **Hanyang**

GitLab login and initial setup

2. Login with Hanyang account



한양대학교 | 로그인


고객님의 정보에 접근하기 위하여 인증이 필요합니다.
한양대학교 포털 한양인(HY-in)계정으로 로그인 하시기 바랍니다.

Portal Login

ID	<input type="text" value="2007002245"/>	<div>로그인</div>
Password	<input type="password" value="....."/>	

GitLab login and initial setup

3. Agree to terms of information provision

 **한양대학교 | 개인정보의 제 3자 제공동의 요청**

한양대학교 OPEN API는 아래와 같은 개인정보를 온라인 소프트웨어 교육 지원 시스템 - Real 에 제공합니다.

제공 받는자

커넥트재단 (온라인 소프트웨어 교육 지원 시스템 - REAL)

제공 목적

웹상에서 학생 실습코드를 저장하고 빌드 하며 채점, 코드 리뷰를 수행하는 시스템 입니다.

실습 코드 저장은 GITLABCE를 활용할 예정이고
코드 리뷰는 REVIEW BOARD, 빌드 및 채점은 JENKINS를 사용할 예정입니다.

한양대 도메인을 통해 서비스 하고 한양대 학생 인증을 적용할 예정입니다.

**** 실제 서비스를 사용자가 사용하는 환경 입니다.**

제공 항목

모든 항목에 동의하셔야만 이용 가능합니다.

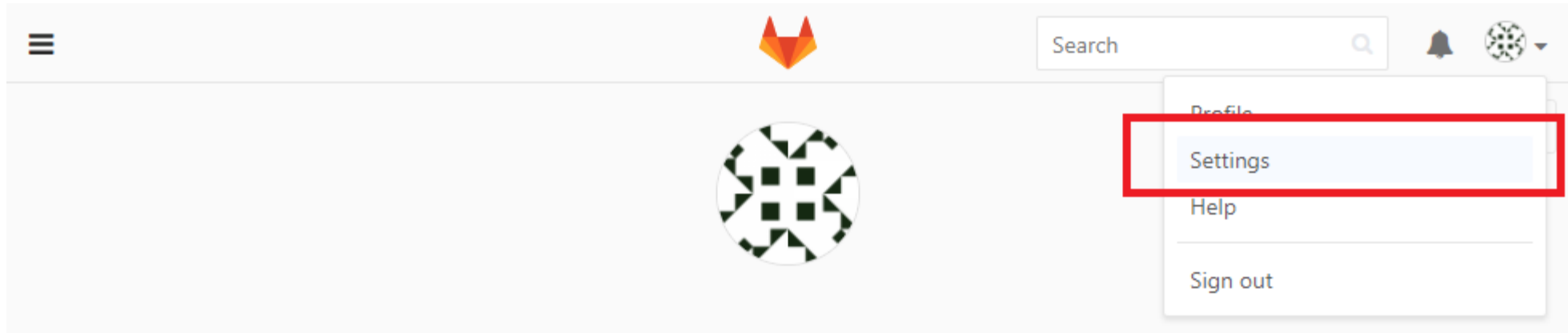
로그인사용자 정보조회

[포털에서 설정한 대표 신분 정보]
로그인한 사용자의 성명, 사용자ID, 학번(개인번호), 재학(재직) 여부, 소속대학, 소속명, 소속코드, 소속ID, 사용자구분명의 정보를 제공합
니다.

☒ 전체 동의합니다. ☐ 동의합니다.

GitLab login and initial setup


4. Set password





GitLab login and initial setup

4. Set password (DO NOT FORGET !!– different password is allowed)

≡ User Settings





[Profile](#) [Account](#) [Applications](#) [Chat](#) [Access Tokens](#) [Emails](#) **[Password](#)** [Notifications](#) [SSH Keys](#) [Preferences](#) [Audit Log](#)

Password

After a successful password update, you will be redirected to the login page where you can log in with your new password.

Change your password or recover your current one

Current password

You must provide your current password in order to change it.

New password




Password confirmation

GitLab login and initial setup

5. Set email

User Settings

Search



Profile

Account

Applications

Chat

Access Tokens

Emails

Password

Notifications


SSH Keys

Preferences

Audit Log

Public Avatar

You can upload an avatar here or change it at gravatar.com



Upload new avatar

Browse file...

No file chosen

The maximum file size allowed is 200KB.

Main settings

This information will appear on your profile.

Name

Enter your name, so people you know can recognize you.

Email

We also use email for avatar detection if no avatar is uploaded.

Update profile settings

Cancel

GitLab login and initial setup

5. Set email – confirm email address in your inbox

GitLab login and initial setup

6. After setting password, you can login without clicking "Sign in with Hanyang"



GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in

Username or email

2007002245

Password

.....|

☐ Remember me

[Forgot your password?](#)

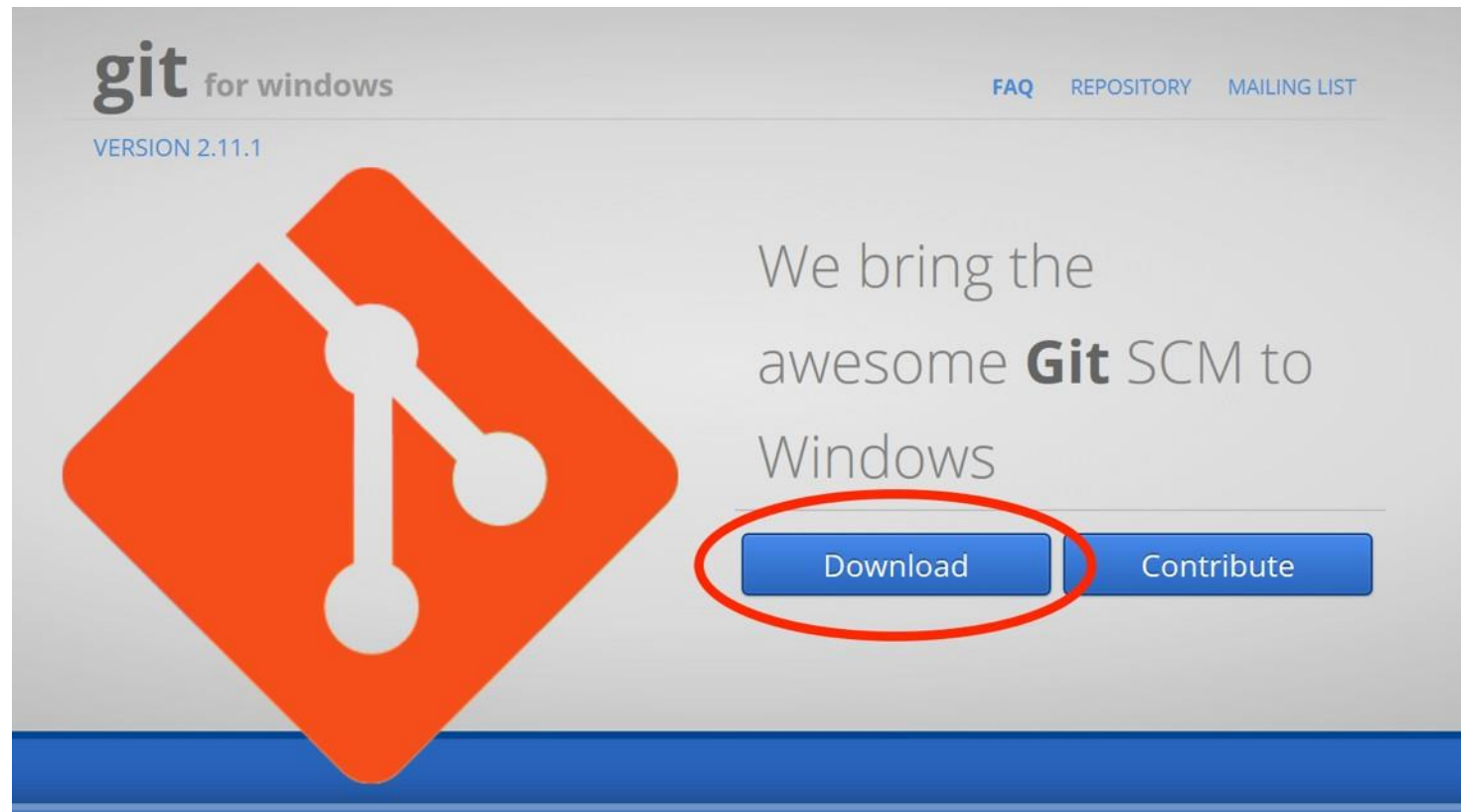
Sign in

Sign in with

Hanyang

Git Installation (Windows)

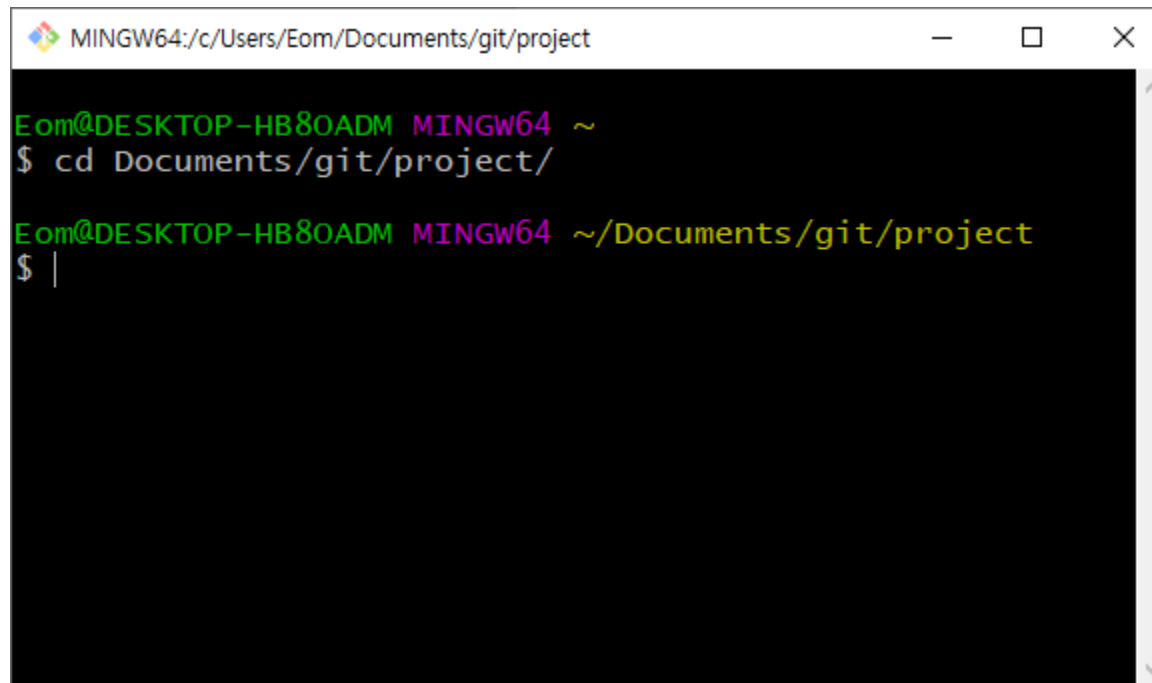
1. <https://git-for-windows.github.io/>



Git Installation (Windows)

2. Launch Git Bash

3. Move to working directory (ex: \$ cd project)



```
MINGW64:/c/Users/Eom/Documents/git/project

Eom@DESKTOP-HB8OADM MINGW64 ~
$ cd Documents/git/project/

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ |
```

Git Installation (Linux)

Ubuntu

```
$ sudo apt-get install git
```

Fedora

```
$ sudo yum install git
```

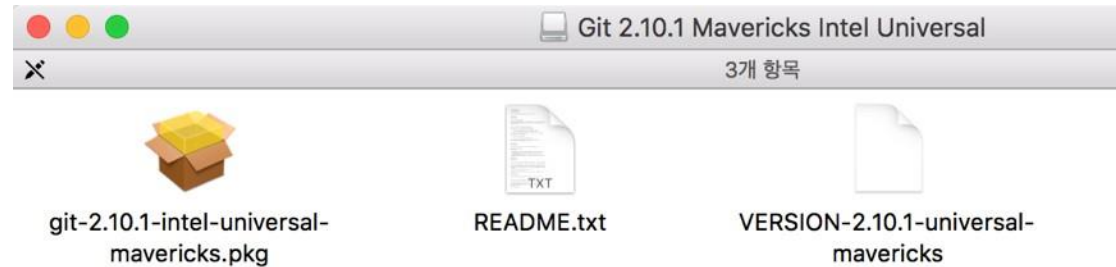


A terminal window titled "mrbin2002 — mrbin2002@ubuntu: ~ — ssh mrbin2002@10.211.55.7 — 74x21". The terminal shows the command `sudo apt-get install git` being executed. The output indicates that git is already the newest version and no packages need to be upgraded. The prompt returns to `mrbin2002@ubuntu:~$`.

```
mrbin2002@ubuntu:~$ sudo apt-get install git
[[sudo] password for mrbin2002:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
mrbin2002@ubuntu:~$
```

Git Installation (MacOS)

1. <https://git-scm.com/download/mac>
2. Mount dmg and launch git-x.x.x-xxx.pkg



Git Basic Usage

1. After installation, set user

```
$ git config --global user.name "student id"  
$ git config --global user.email "student id @hanyang.ac.kr"
```

(user.name is student id,
user.email is email registered Gitlab(<https://hconnect.hanyang.ac.kr>)

Git Basic Usage

2. Clone your Git repository

```
$ git clone https://hconnect.hanyang.ac.kr/2020_ITE2031_11813/2020_ITE2031_XXXXXXXXXX.git
```

2020_ITE2031_11813 / 2020_ITE2031_XXXXXXXXXX

Project Activity Pipelines Issues 0 Merge Requests 0 Wiki

2

2020_ITE2031_20

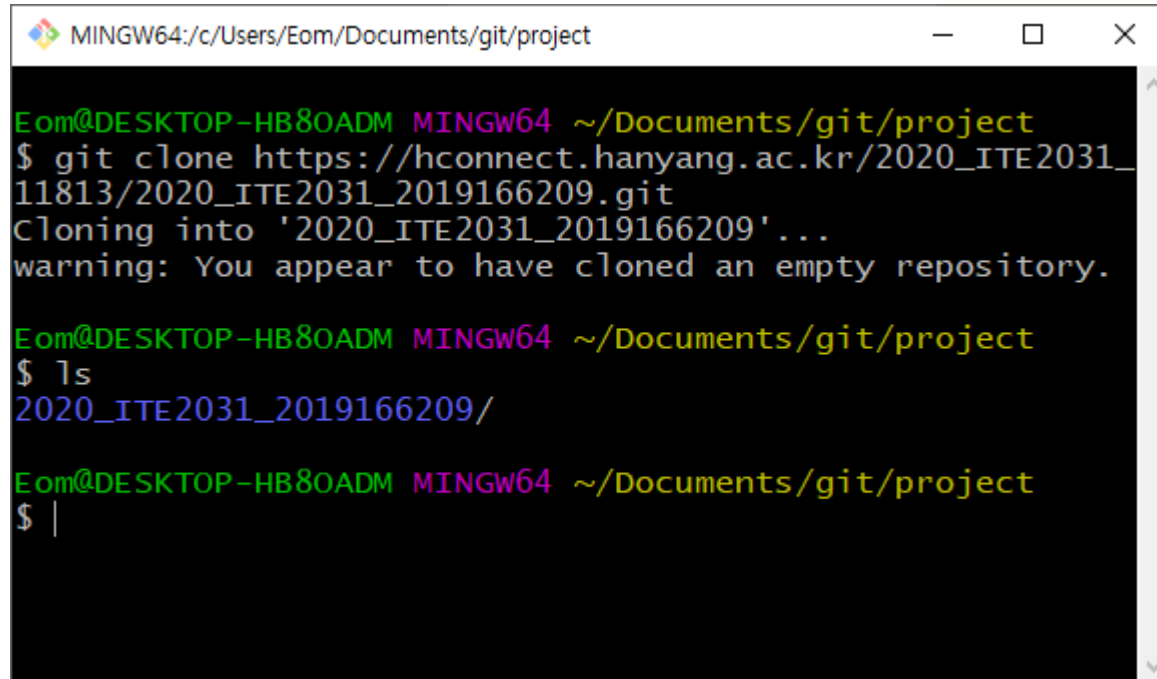
컴퓨터 소프트웨어학과, 컴퓨터구조론, 박영준 교수님, 월 14:30~16:00(IT.BT 508), 화 14:30-16:00(IT.BT 508), \Department of Computer Science, Computer Architecture, Yongjun Park, Mon 14:30-16:00(IT.BT 508), Tue 14:30-16:00(IT.BT 508)

Star 0 HTTPS https://hconnect.hanyang.ac.kr/2020_ITE2031_11813/2020_ITE2031_XXXXXXXXXX.git + - Global

Copy & Paste

Git Basic Usage

3. Enter username(student id) and password(set in GitLab)

A screenshot of a terminal window titled 'MINGW64:/c:/Users/Eom/Documents/git/project'. The terminal shows the following commands and output:

```
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ git clone https://hconnect.hanyang.ac.kr/2020_ITE2031_11813/2020_ITE2031_2019166209.git
Cloning into '2020_ITE2031_2019166209'...
warning: You appear to have cloned an empty repository.

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ ls
2020_ITE2031_2019166209/

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ |
```

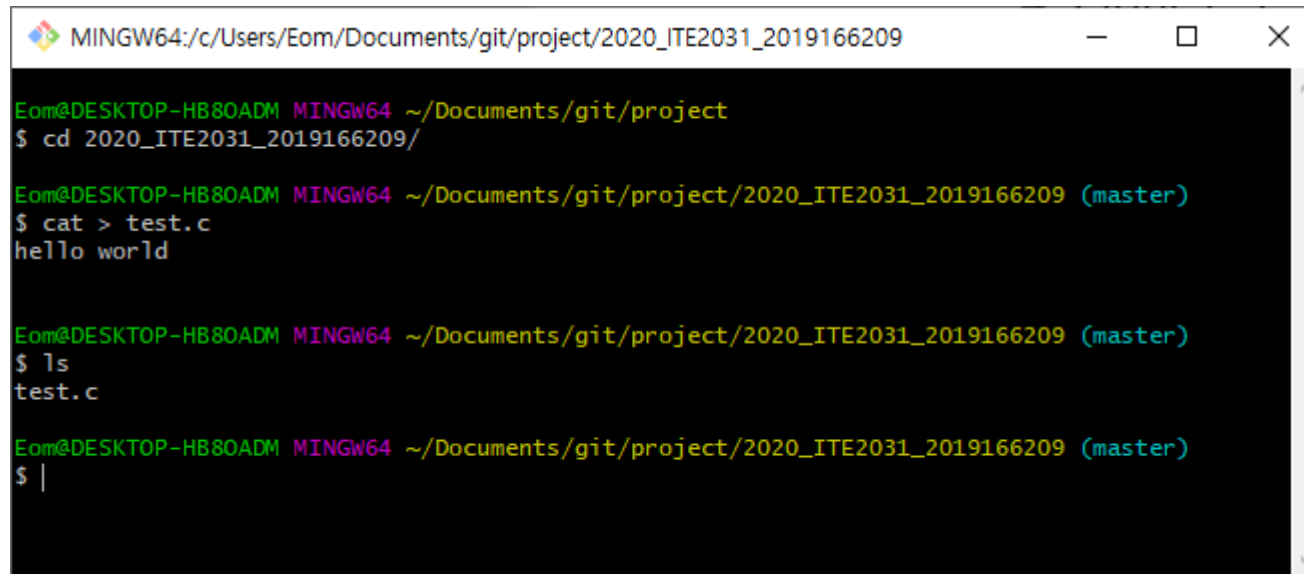

Git Basic Usage

4. Move to cloned directory

```
$ cd 2020_ITE2031_XXXXXXXXXX
```

5. Create file

```
$ vi test.c
```



```
MINGW64:/c:/Users/Eom/Documents/git/project/2020_ITE2031_2019166209
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ cd 2020_ITE2031_2019166209/

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ cat > test.c
hello world

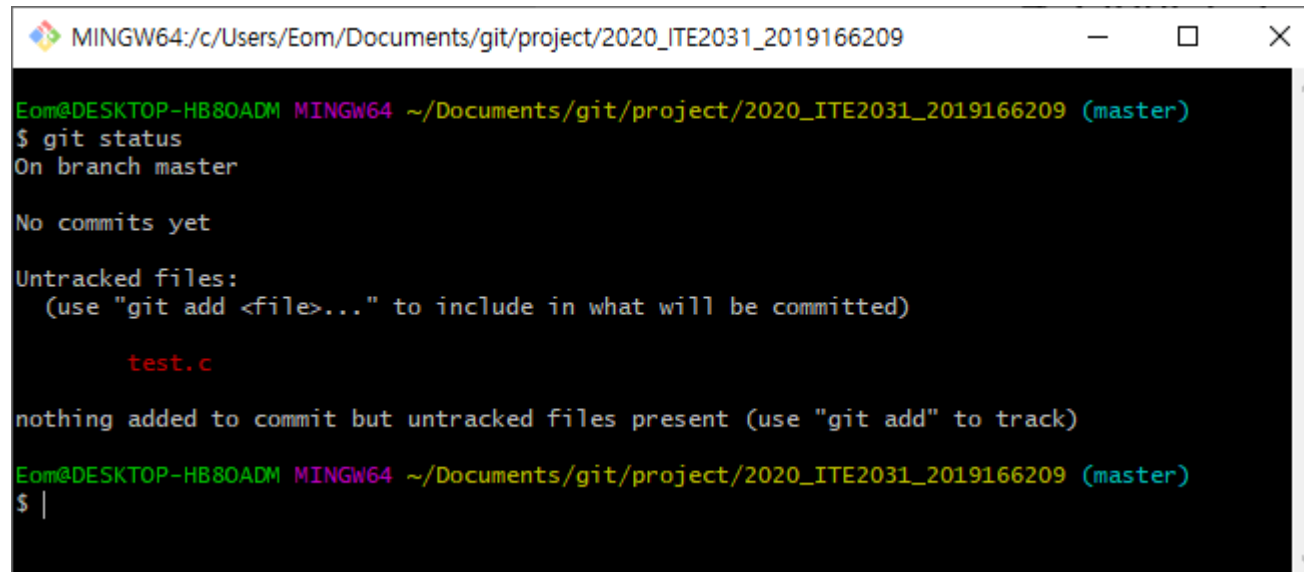
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ ls
test.c

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ |
```

Git Basic Usage

6. If you check current status, created file is categorized as untracked.

```
$ git status
```



```
MINGW64:/c:/Users/Eom/Documents/git/project/2020 ITE2031_2019166209
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020 ITE2031_2019166209 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.c

nothing added to commit but untracked files present (use "git add" to track)
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020 ITE2031_2019166209 (master)
$ |
```

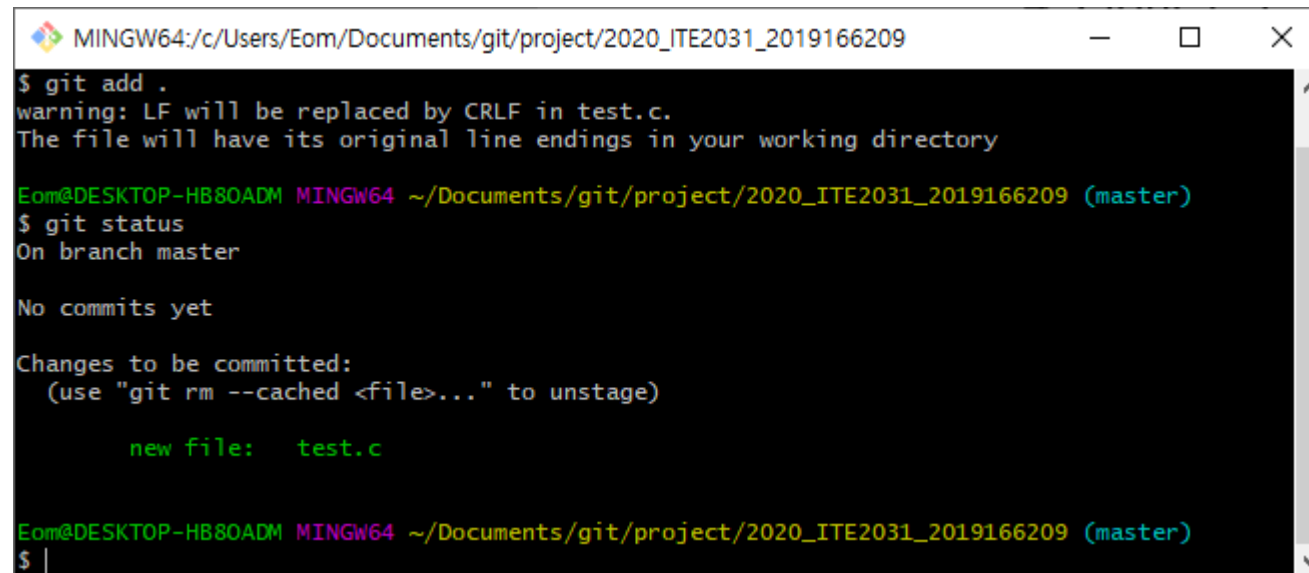
Git Basic Usage

7. Move all created or modified files in directory to staged area

```
$ git add .
```

8. Check status again

```
$ git status
```



```
MINGW64:/c:/Users/Eom/Documents/git/project/2020_ITE2031_2019166209
$ git add .
warning: LF will be replaced by CRLF in test.c.
The file will have its original line endings in your working directory

Eom@DESKTOP-HB80ADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.c

Eom@DESKTOP-HB80ADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$
```

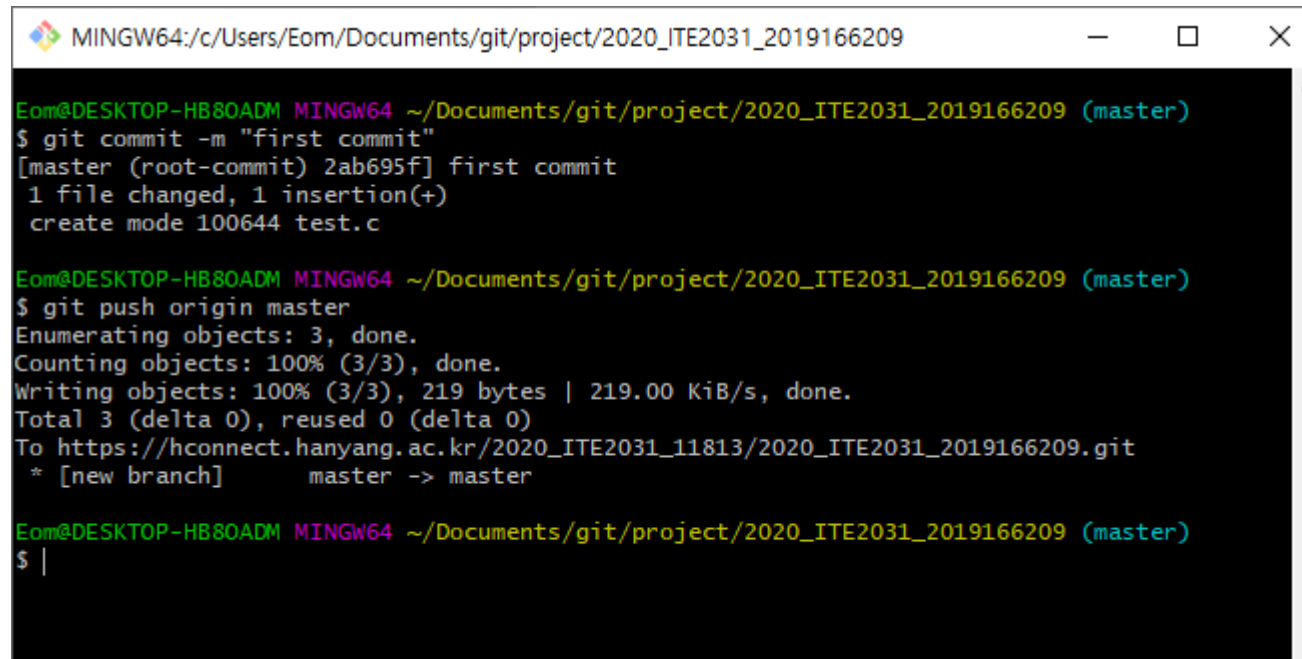
Git Basic Usage

9. Commit added or modified files(affect only local repository)

```
$ git commit -m "first commit"
```

10. Push commits to remote repository

```
$ git push origin master
```



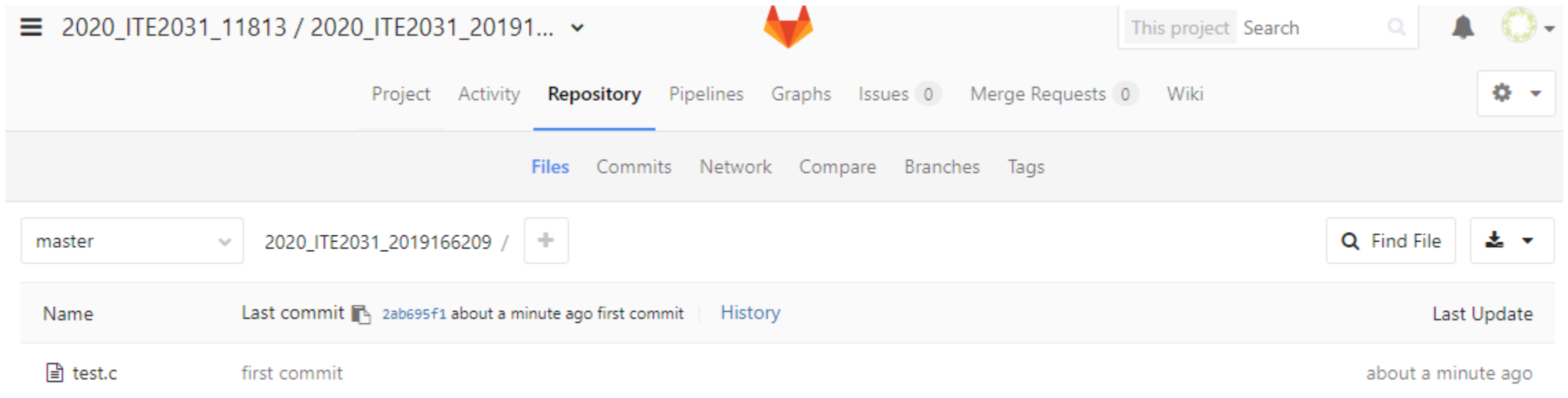
```
MINGW64:/c/Users/Eom/Documents/git/project/2020_ITE2031_2019166209
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ git commit -m "first commit"
[master (root-commit) 2ab695f] first commit
1 file changed, 1 insertion(+)
create mode 100644 test.c

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://hconnect.hanyang.ac.kr/2020_ITE2031_11813/2020_ITE2031_2019166209.git
 * [new branch]      master -> master



Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ |
```

Git Basic Usage

11. Commits sent to the remote repository via git push are visible in the GitLab webpage



The screenshot displays the GitLab web interface for a repository named '2020_ITE2031_11813 / 2020_ITE2031_20191...'. The 'Repository' tab is selected, showing the 'Files' view. The current branch is 'master'. The repository path is '2020_ITE2031_2019166209 /'. A search bar labeled 'Find File' and a download icon are visible. Below the navigation bar, a table lists the files in the repository:

Name	Last commit  2ab695f1 about a minute ago first commit History	Last Update
 test.c	first commit	about a minute ago