

---

# **Creative Software Programming**

## **Lab1: Environment Setting, Git / Gitlab, Vim**

Yoonsang Lee

Fall 2019

# Topic Covered

---

- Install Ubuntu
- Git / Gitlab
- Vim Basic Usage

# Development Environment

---

- (If you're using OS other than Ubuntu) Use virtual machine: <http://www.virtualbox.org/>
- Ubuntu: Ubuntu 18.04 is recommended.
  - <http://releases.ubuntu.com/18.04>
- Editor: Vim is recommended.

---

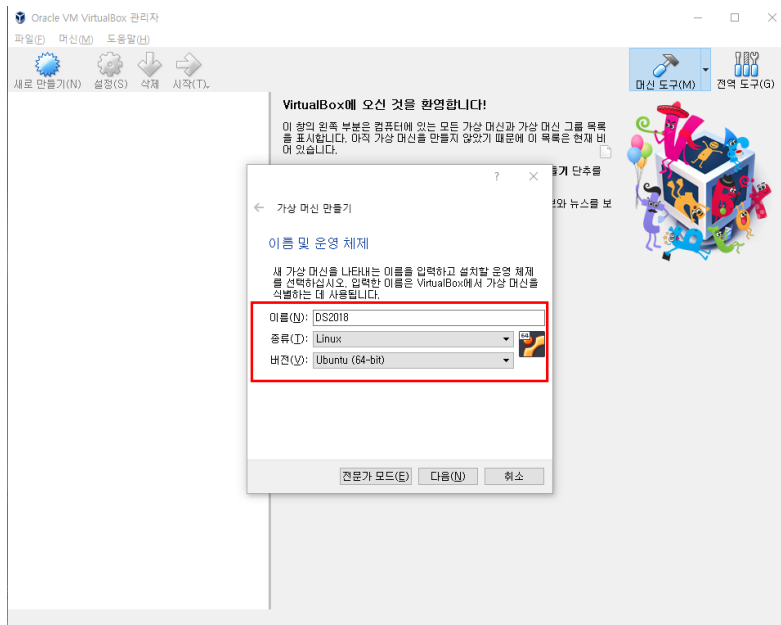
# **Install Ubuntu**

# Install Ubuntu in Virtual Box

---

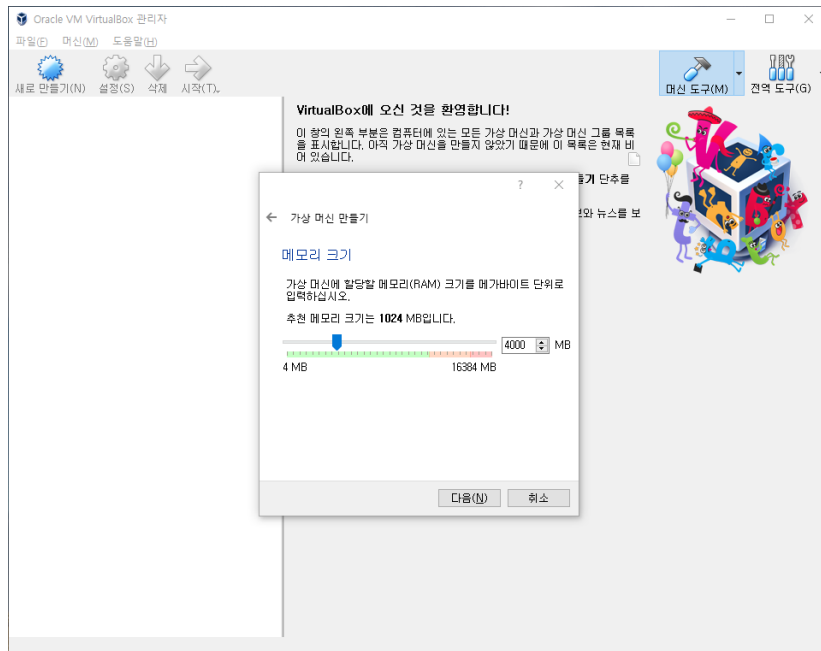
- If you're using a computer with Ubuntu installed, you can use it as is.
- Following slides assume you're using other OSs.

# How to install Ubuntu



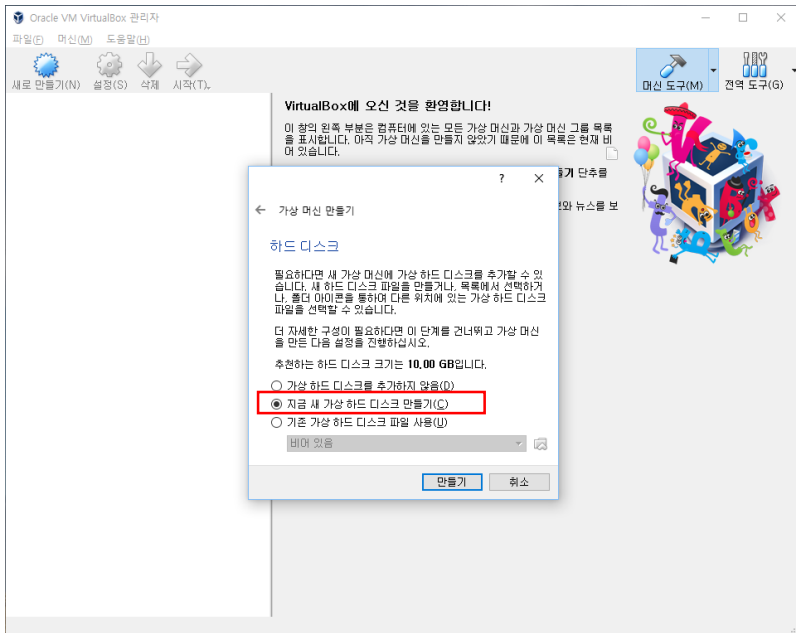
- Name : (any name you want)
- Type : Linux
- Version : Ubuntu (64-bit)

# How to install Ubuntu



- Memory size : (any size)

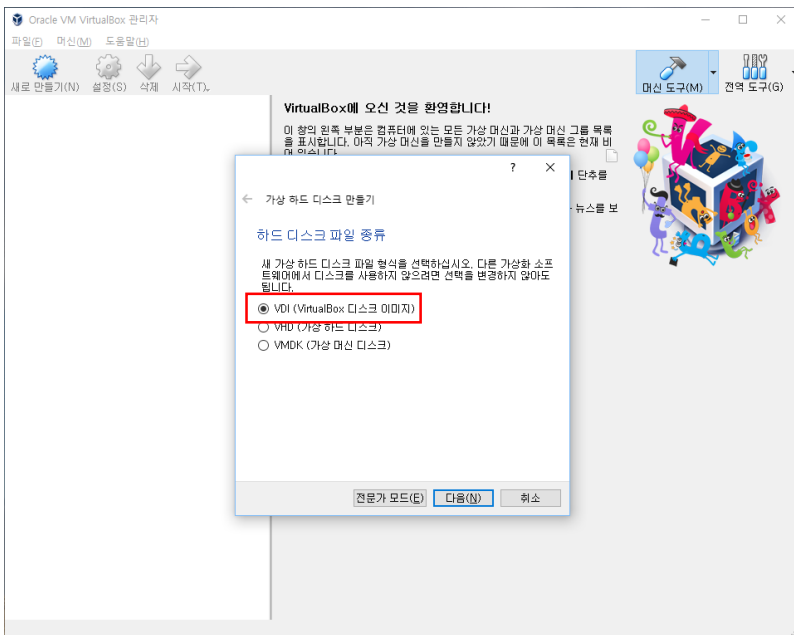
# How to install Ubuntu



- Create a virtual hard disk

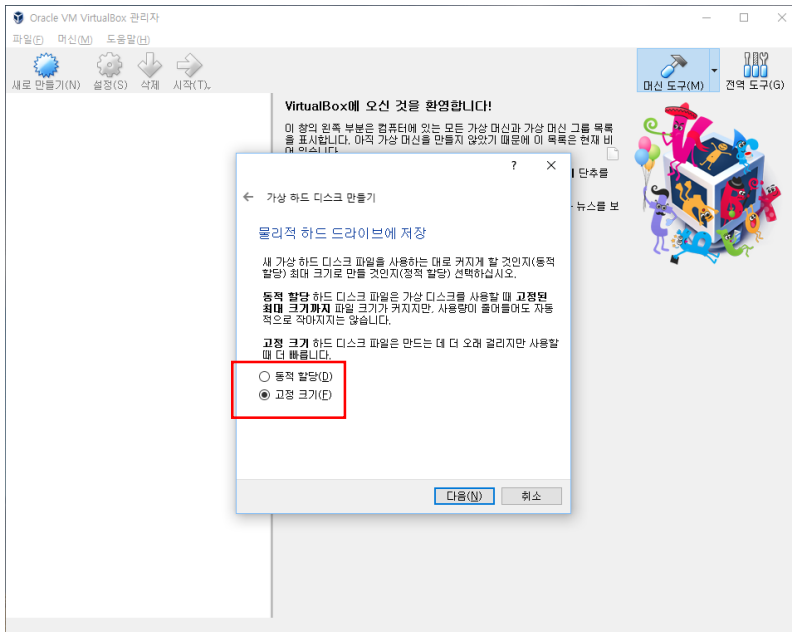


# How to install Ubuntu



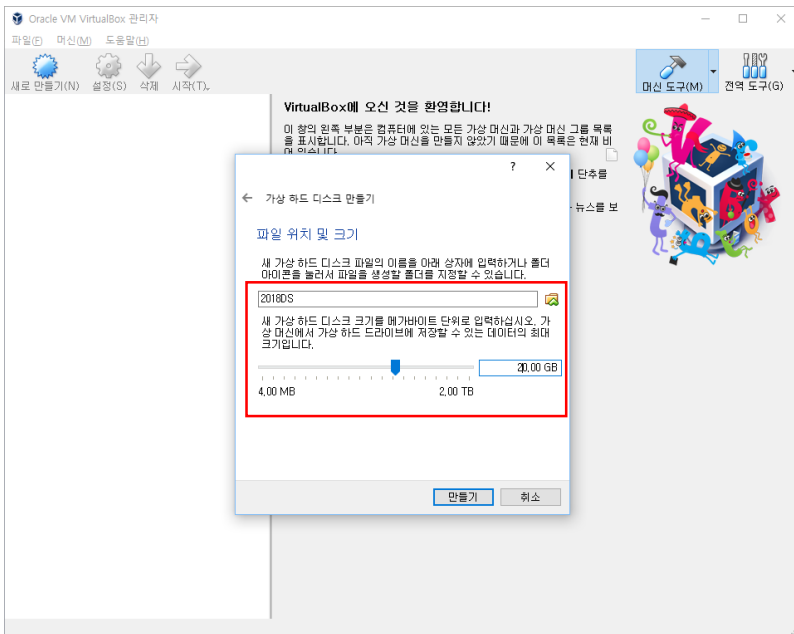
- VDI

# How to install Ubuntu



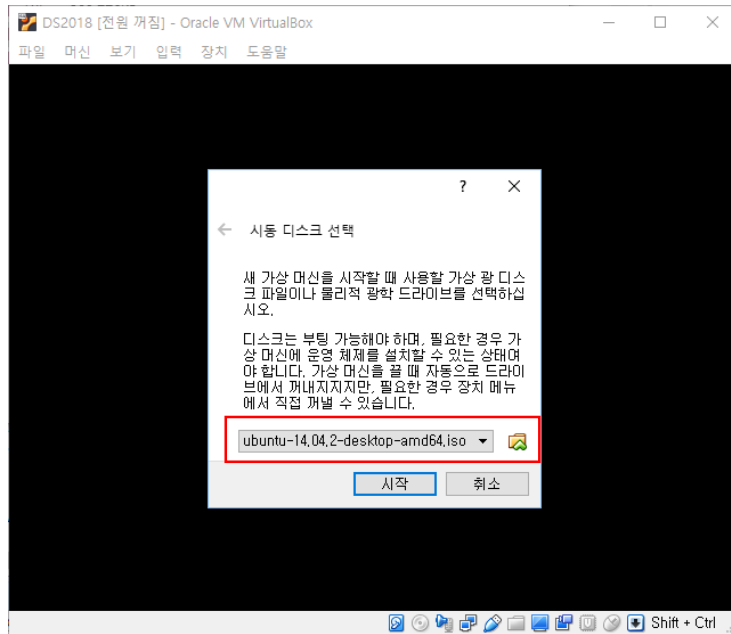
- Recommendation: Fixed size

# How to install Ubuntu



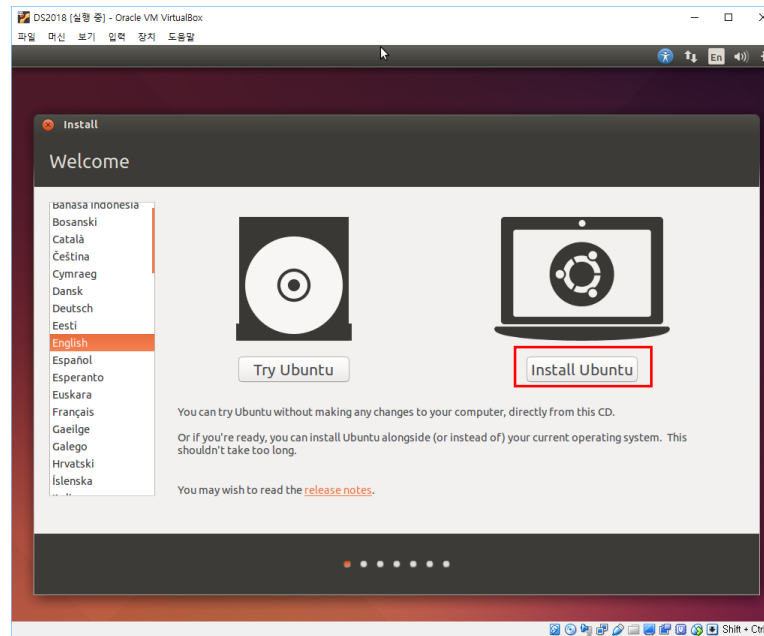
- Virtual disk file location & size:
- any location you want
- any size you want (e.g. 20GB)

# How to install Ubuntu

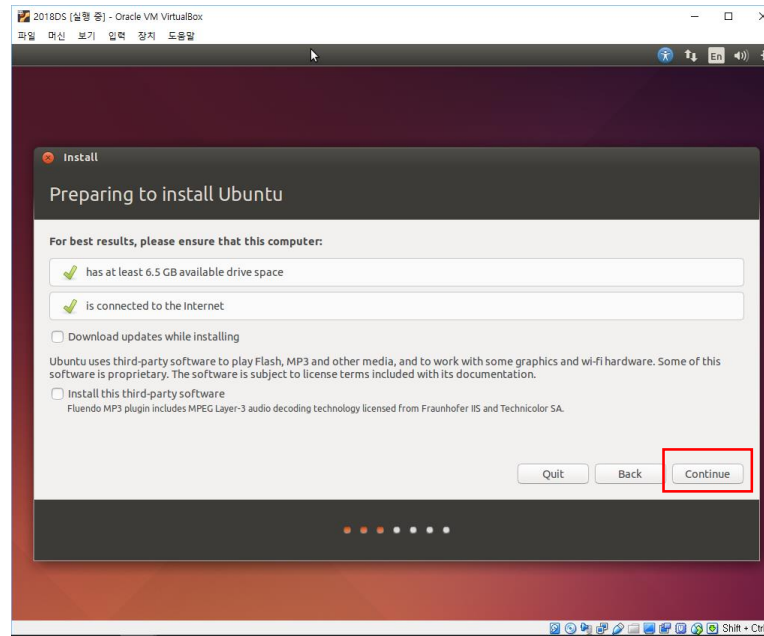


- Choose the downloaded Ubuntu .iso file as a boot disk

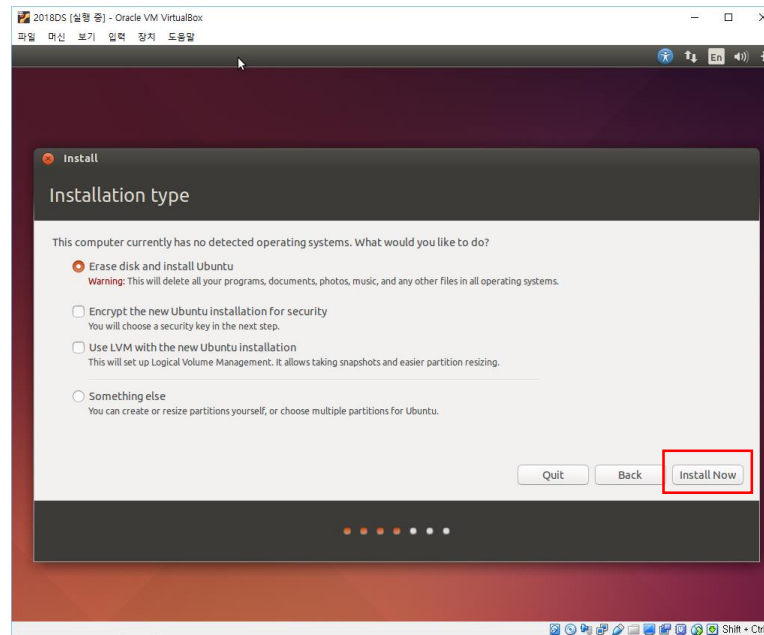
# How to install Ubuntu



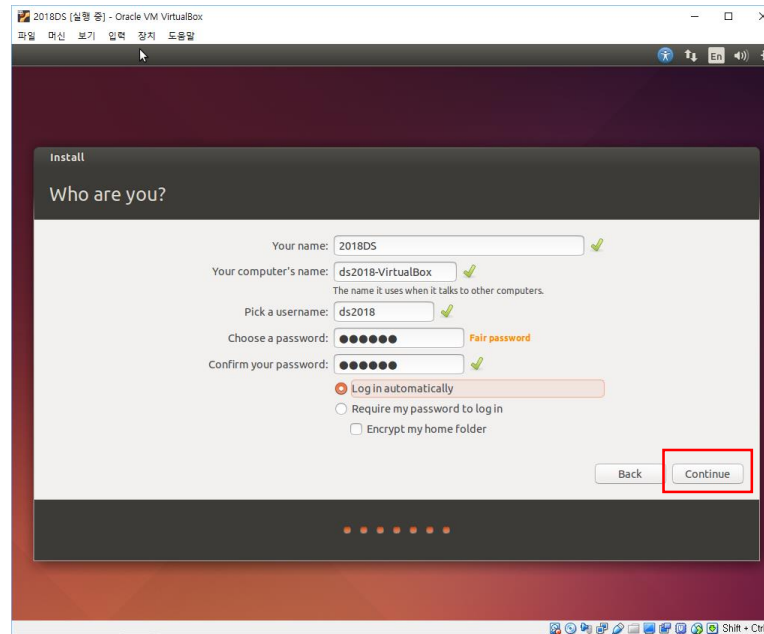
# How to install Ubuntu



# How to install Ubuntu



# How to install Ubuntu





---

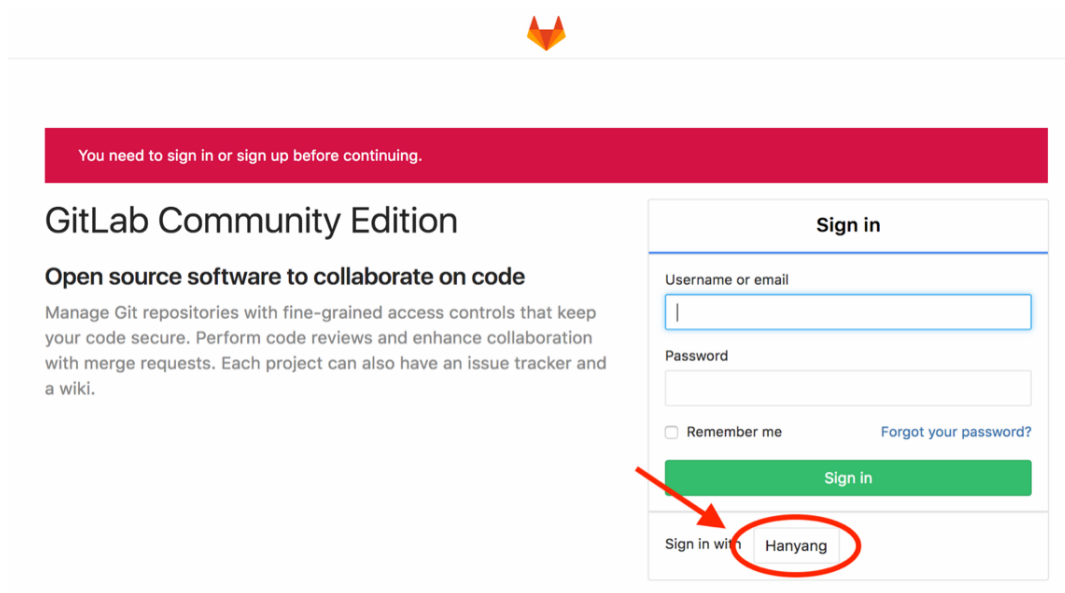
# **Git / Gitlab**

# Git

- In this course, you should submit your assignments on the gitlab server of our department:
- <https://hconnect.hanyang.ac.kr>

# Git

- Access to <https://hconnect.hanyang.ac.kr/>



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**

# Git

- Login hanyang account



## 한양대학교 | 로그인

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

Portal Login

---

ID

Password

**로그인**

# Git

- Consent for 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 하여 채점, 코드 리뷰를 수행하는 시스템입니다.

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 동의하시어만 이용 가능합니다.

**로그인사용자 정보조회**

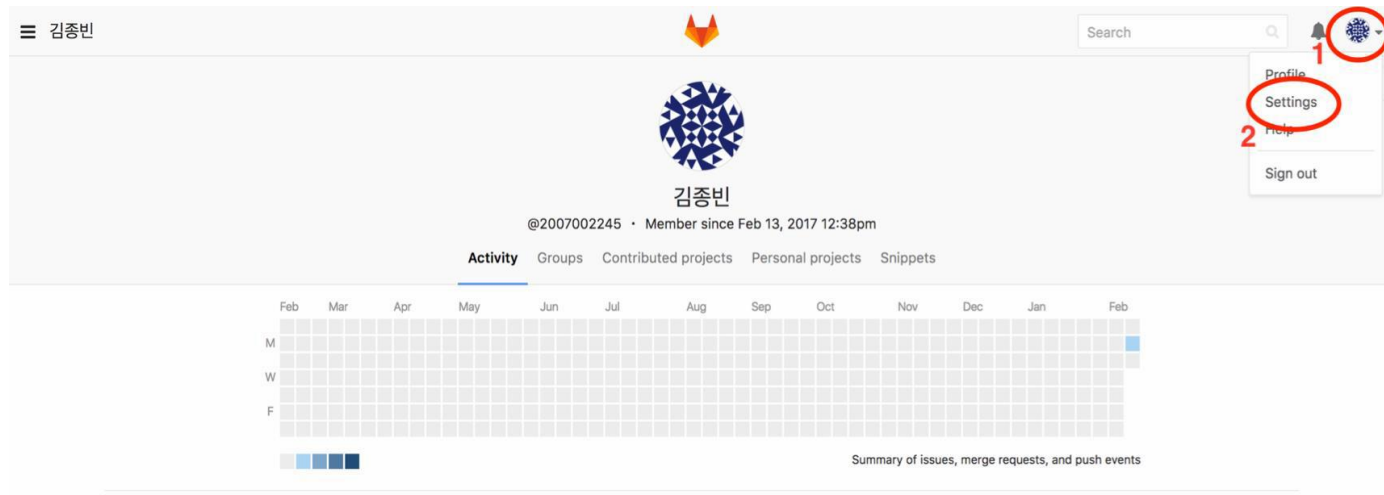
[포털에서 설정한 대표 신분 정보]  
로그인한 사용자의 성명, 사용자ID, 학번(개인번호), 재학(재직) 여부, 소속대학, 소속명, 소속코드, 소속ID, 사용자구분명의 정보를 제공합니다.

전체 동의합니다.  동의합니다.

동의  취소

# Git

- Set up Password



## • Set up Password

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  initial passwd: HY-in passwd

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

New password

Password confirmation

**Save password** [forgot my password](#)

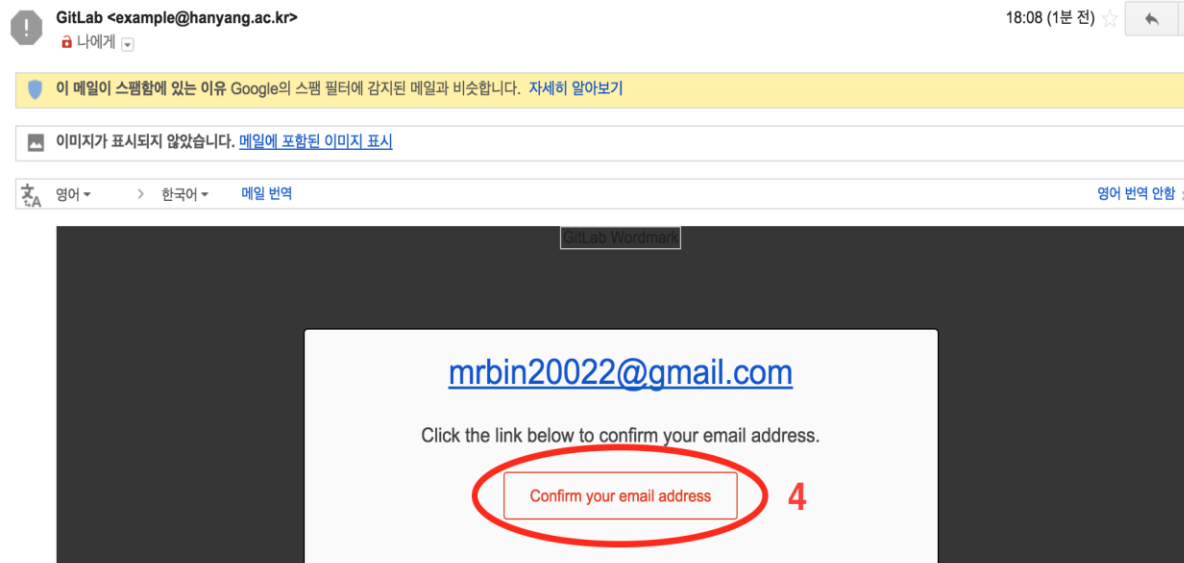
- Set up Email

The screenshot shows the GitHub User Settings page. The 'Profile' tab is selected and circled with a red circle and the number '1'. Below the navigation bar, the 'Public Avatar' section shows a green and white checkered avatar. The 'Main settings' section contains three input fields: 'Name' (filled with '김종빈'), 'Email' (filled with 'mrbin20022@gmail.com' and circled with a red circle and the number '2'), and 'Bio' (empty). At the bottom, the 'Update profile settings' button is circled with a red circle and the number '3'. A 'Cancel' button is also visible.



# Git

- Set up Email – Approve from changed email



# Git

- After this, you can sign in with student ID / email and changed password (Without going through 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

Password

Remember me [Forgot your password?](#)

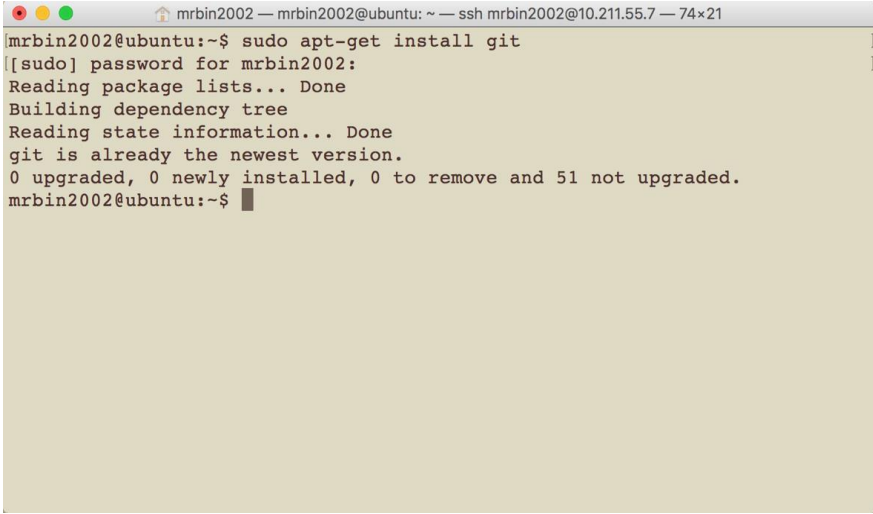
Sign in with

# Git

- Install Git (Linux)

Ubuntu

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



```
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

- Set up Git user info

```
$ git config --global user.name "2007002245"
```

```
$ git config --global user.email "2007002245@hanyang.ac.kr"
```

( user.name is your student\_ID

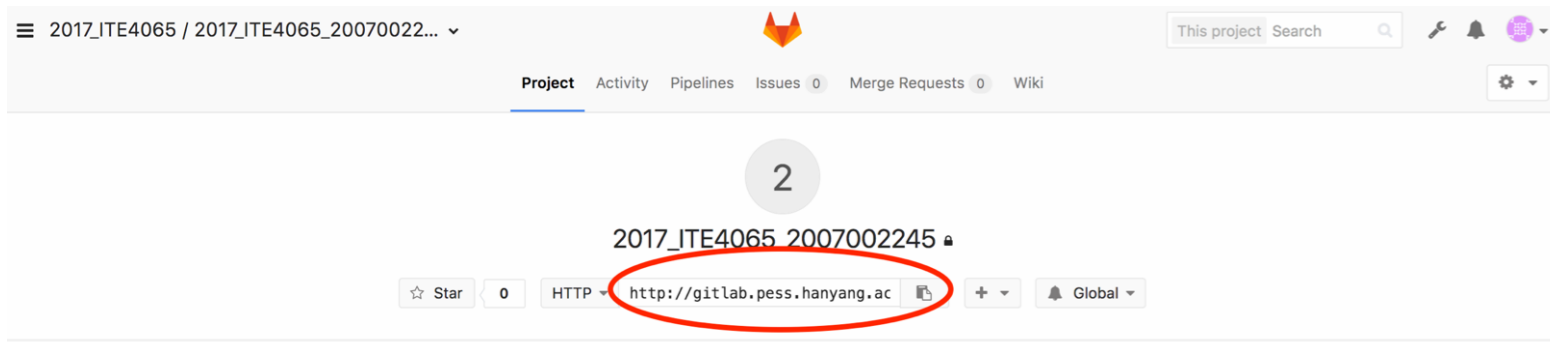
user.email is registered Email on GitLab (Default: student\_ID + @hanyang.ac.kr))

# Git

- Clone Git Repository

```
$ git clone https://hconnect.hanyang.ac.kr/2019_CSE4020_12030/2019_CSE4020_StudentID.git
```

You can check your Git URL in GitLab project page



# Git

- When you clone your project
- Username : StudentID
- Password : Password registered in GitLab(page 7)

```
TA — mrbin2002@ubuntu: ~ — -bash — 76x21
Jongbin:TA mrbin2002$ git clone http://gitlab.pess.hanyang.ac.kr/2017_ITE4065/2017_ITE4065_2007002245.git
Cloning into '2017_ITE4065_2007002245'...
Username for 'http://gitlab.pess.hanyang.ac.kr': 2007002245
Password for 'http://2007002245@gitlab.pess.hanyang.ac.kr':
warning: You appear to have cloned an empty repository.
Jongbin:TA mrbin2002$ ls
2017_ITE4065_2007002245
Jongbin:TA mrbin2002$ █
```

# Git

- Move to Clone directory

```
$ cd YEAR_CSE4020_Student ID
```

- Make test file

```
$ vi test.c
```



A screenshot of a terminal window. The title bar shows the window name as '2017 ITE4065\_2007002245 — mrbin2002@ubuntu: ~ — vi test.c — 76x21'. The terminal content shows a file editor with the text '1 hello world' on the first line, followed by three lines of tilde characters (~) representing empty lines.

# Git

- When you check the current git status, “test.c” is displayed as untracked.

```
$ git status
```

```
[Jongbin:2017_ITE4065_2007002245 mrbin2002$ git status  
On branch master  
  
Initial commit  
  
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)  
Jongbin:2017_ITE4065_2007002245 mrbin2002$
```



# Git

- Move all added/modified files in the current directory to Stage area (It means that test.c file is managed by Git)

```
$ git add .
```

- Check git status

```
$ git status
```

```
[Jongbin:2017_ITE4065_2007002245 mrbin2002$ git add .
[Jongbin:2017_ITE4065_2007002245 mrbin2002$ git status
On branch master

Initial commit

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

    new file:   test.c
```

# Git

- Commit added/modified files (Save on Local repository)

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

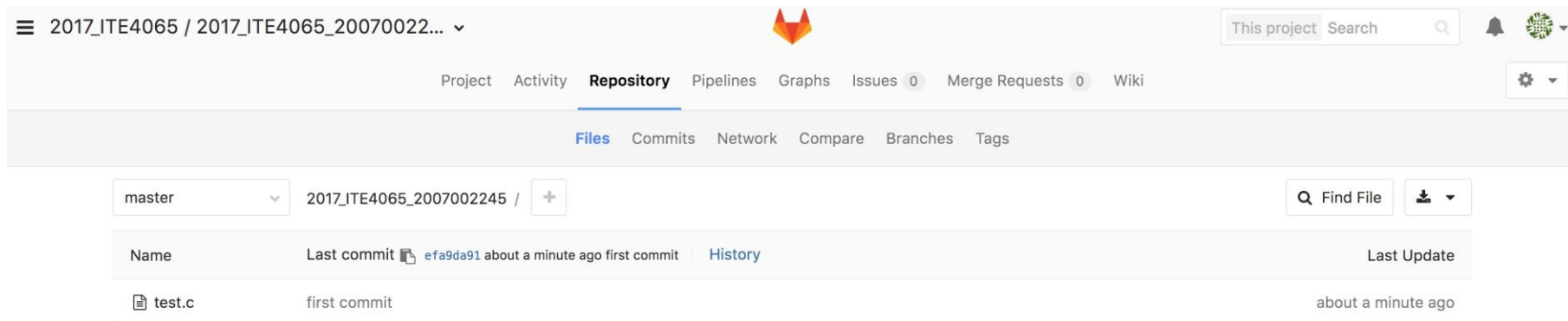
- Send Committed Content to Remote Repository

```
$ git push origin master
```

```
Jongbin:2017_ITE4065_2007002245 mrbin2002$ git commit -m "first commit"
[master (root-commit) efa9da9] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.c
Jongbin:2017_ITE4065_2007002245 mrbin2002$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 224 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To http://gitlab.pess.hanyang.ac.kr/2017_ITE4065/2017_ITE4065_2007002245.git
 * [new branch]      master -> master
```

# Git

- Files sent to the remote through the git push can be found on the GitLab web page.



The screenshot shows a GitLab repository page for a project named '2017\_ITE4065'. The repository is on the 'master' branch. A file named 'test.c' is listed with its last commit being 'first commit' by user 'efa9da91' about a minute ago. The page includes navigation tabs for Project, Activity, Repository, Pipelines, Graphs, Issues (0), Merge Requests (0), and Wiki. The 'Files' tab is active, showing a table of files.

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

---

# **Vim Basic Usage**

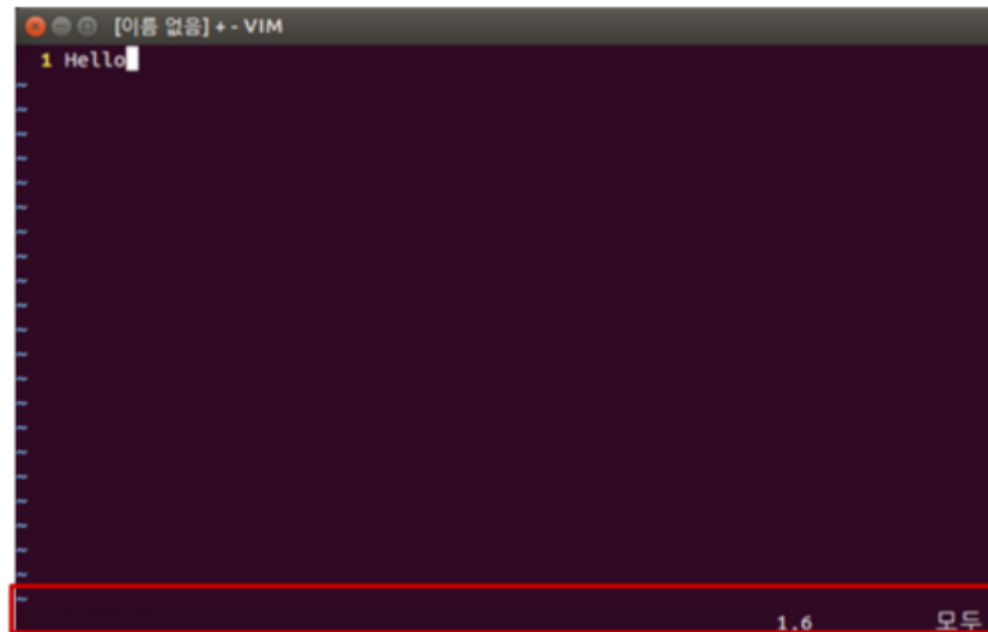
# Vim



- **Vi IMproved**
- An editor with many improvements to the existing Vi editor (first released in 1991 by Bram Moolenaar)
  - Vi is created in 1976 by Bill Joy, a key developer of BSD
  - Vim is used much more now.
- Vim is a default editor in most Linux systems.
  - Knowing how to use Vim is a great way to work on Linux.
- Three modes in Vim:
  - Normal mode
  - Insert mode
  - Command-line mode

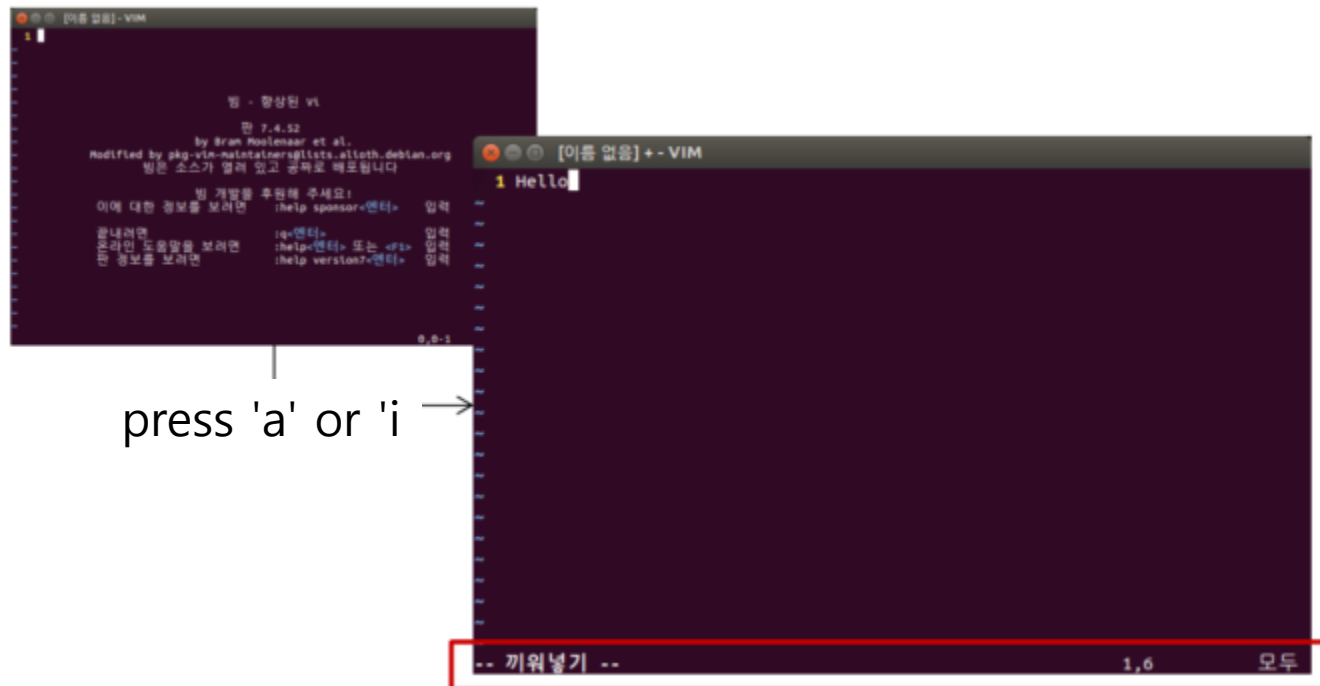
# Normal Mode

- Vim starts in Normal mode.
- Copy, paste, delete, search and other functions are available through shortcut keys.
- In this mode, vim is waiting for your command shortcut.
- Press **ESC** in other modes to enter Normal mode.



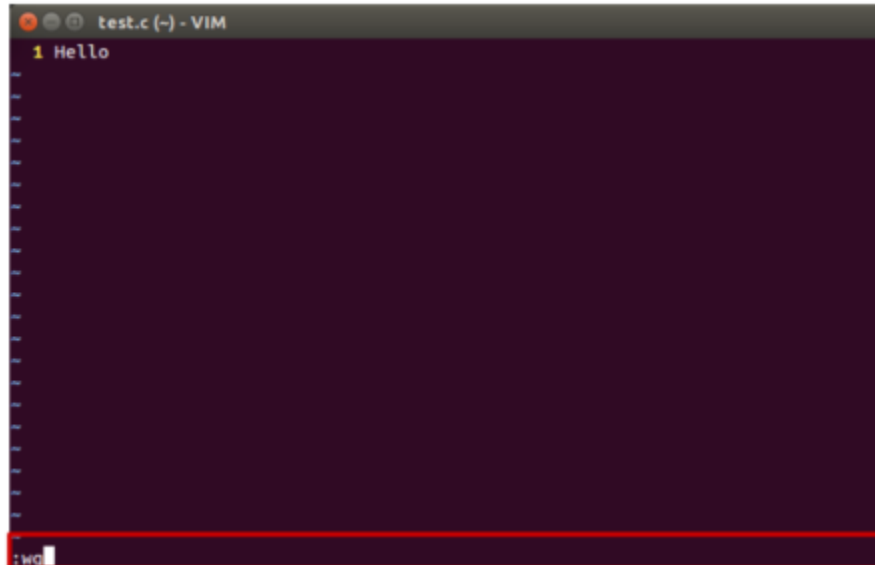
# Insert Mode

- Press **a** or **i** in Normal mode to enter Insert mode.
- In this mode, you can enter and edit a file as you would in a general text editor.



# Command-line Mode

- Press `:` in Normal mode to enter Command-line mode.
- In this mode, you can enter commands on the command line in vim.
  - `w` : save
  - `q` : quit
  - `!` : "force" something (ex : `wq!` , `q!`: force save, force quit)



The screenshot shows a terminal window titled "test.c (-) - VIM". The main area of the window contains a single line of text: "1 Hello". At the bottom of the window, a red horizontal bar indicates the command-line mode, where the text ":w" is visible, followed by a cursor. This represents the user entering the 'w' command to save the file.



# References for Vim Basic Usage

---

- Vimtutor: A tutorial included in vim

(Shell)

```
vimtutor
```

- Additional tutorials:
  - Interactive Vim tutorial  
<http://www.openvim.com/tutorial.html>

# References for Vim Commands

---

- Vim Cheat Sheet : <https://vim.rtorr.com/lang/ko/>
- `:help <command>` : help document for the command

# Vim Cursor Movement Commands

previous  
**gg**  
first line

**#** **n** **?text** **N**  
find word under cursor previous text find text next text

**C-b** **C-u**  
page 1/2 page

**H**  
screen

**{** **(**  
paragraph sentence

**;** **B** **gE** **,**  
previous x delimited word delimited end previous x

**O** **^** **Fx** **Tx** **b** **ge** **h** **k** **l** **e** **w** **tx** **fx** **\$**  
line non-blank find x after x word end left up right end word before x find x line

**,** **)** **}** **E** **W** **;**  
next x sentence paragraph delimited end delimited word next x

**L**  
screen

**C-d** **C-f**  
1/2 page page

**N** **/text** **n** **\***  
previous text find text next text find word under cursor

**G**  
last line next

back

forward

absolute movements

**''** **'.** **#G** **%**  
last location last edit line # matching bracket



# Vim Advanced Usage

---

- Shell settings for convenient vim use
- `.vimrc` - vim configuration file
- Vim visual mode
- Vim windows
- Vim plug-ins
  
- For details, see the supplementary material: *1-Lab1-VimAdvanced.pdf*

# Assignment 1-1

---

- Now it's time for the first assignment.
- Assignment 1-1 is only for practice and will not be included in the final grade.
- However, you must complete and submit the assignment to understand how to submit the assignment.
- Course home (at [learn.hanyang.ac.kr](http://learn.hanyang.ac.kr)) - Assignments - Assignment 1-1