
Computer Graphics

1 - Lab: Environment Setting for Lectures & Assignments

Yoonsang Lee
Spring 2020

Introduction

- TA: Geuntae Park (박근태)
 - qkrrmsxo01@hanyang.ac.kr
- Undergraduate Mentor: Jong Woon Baek (백종운)

Topics Covered

- Installing Python Interpreter
- Python 2 & Python 3
- Installing Additional Python Modules
- Running Python Interpreter
- Creating a Gitlab account
- Time for Lab Assignment 1

Install Python Interpreter

- Python **3.7** or later
 - <https://www.python.org/downloads/>
- Note that all submissions for assignments should work in Python **3.7**.
- You can use any OS that runs Python.

Python 2 & Python 3

- Python 2 is still in active use.
- Python 3 is the future of Python.
 - A lot of very useful features & fixes for well-known problems
 - To do this, **Python 3 breaks backward compatibility**.
- If you're familiar with Python 2, you have to know the difference between Python 2 and 3.
 - The following link would be helpful:
 - http://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html

Install Python Packages (Modules)

- My recommendation for installing python modules is using **pip** (Python Package Index).
 - pip is a program that helps you install most of the python package.
 - pip is already installed if you are using Python 2 $\geq 2.7.9$ or Python 3 ≥ 3.4 downloaded from python.org
- Usage:

```
pip install <package_name>
```

Python Virtual Environments

- Example: Two python projects on the same machine,
 - Project A are based on Django 1.11
 - Project B are based on Django 2.1.7
 - Python interpreter cannot differentiate between versions of Django packages! (So you can run only one of them)
- Python virtual environment:
 - A self-contained directory tree that contains a Python installation for a particular version of Python, plus a number of additional packages.
 - You can keep dependencies required by different projects separate by creating "isolated" python virtual environments for them.

Python Virtual Environments

- **It is generally good to have one virtual environment for each Python project you work on.**
 - So the dependencies of every project are isolated from the system and each other.
- Two most popular tools:
 - virtualenv
 - Anaconda

Install virtualenv & virtualenvwrapper

- Windows

```
> pip install virtualenv virtualenvwrapper-win  
(or)  
> py -3 -m pip install virtualenv virtualenvwrapper-win
```

- Ubuntu

```
# if you don't have pip, install it first.  
$ sudo apt-get install python3-pip  
  
$ sudo pip install virtualenv virtualenvwrapper  
  
# Add the following lines to ~/.bashrc:  
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3  
source /usr/local/bin/virtualenvwrapper.sh  
  
$ source ~/.bashrc
```

(You can skip this process if you're already using virtualenv or Anaconda.)

Install virtualenv & virtualenvwrapper

- MacOS

```
# Install Homebrew(package manager for mac OS) from  
below link.
```

```
https://brew.sh/index\_ko
```

```
# if you install python3 using Homebrew, pip and pip3  
would be installed automatically.
```

```
$ brew install python3
```

```
$ pip3 install virtualenv virtualenvwrapper
```

```
# Add the following line to ~/.bashrc:
```

```
export VIRTUALENVWRAPPER_PYTHON=/usr/local/bin/python3  
source /usr/local/bin/virtualenvwrapper.sh
```

```
$ source ~/.bashrc
```

How to use virtualenvwrapper

```
# Create an environment
$ mkvirtualenv --python=PATH_TO_PYTHON ENVNAME

# Remove an environment
$ rmvirtualenv ENVNAME

# List all of the environments
$ lsvirtualenv

# Activate an environment
$ workon ENVNAME

# Deactivate the current environment
$ deactivate
```

Create an environment for this class

- Windows

```
> mkvirtualenv --python=<python_path> cg-course
```

- An example for <python_path>:

"C:\Users\<your_id>\AppData\Local\Programs\Python\Python35\python.exe"

- If your system does not know “mkvirtualenv”, you need to add python script directory (e.g. ...\\Python35\\Script\\) to system path.

- Ubuntu, MacOS

```
> mkvirtualenv --python=<python3.x> cg-course
```

- Replace <python3.x> with your python version

- e.g. --python=python3.7

Activate the environment

```
$ workon cg-course
```

- Then you can see the name of your environment in the command prompt.
- You can run the exact version of python interpreter specified in the environment just by typing "python".

Install Additional Modules

- We'll use a few python modules in this class
 - NumPy, PyOpenGL, glfw
- In the "cg-course" environment by activating it,

```
$ workon cg-course
```

- NumPy
 - Windows, Ubuntu, MacOS

```
$ pip install numpy
```

Install Additional Modules

- PyOpenGL

- Windows

```
> pip install pipwin  
> pipwin install pyopengl
```

- or, you can also try the "pip install pyopengl" route for windows. But in that case the pyopengl installer wouldn't install the GLUT library properly. You can manually install GLUT and copy glut32.dll to "python-installation-directory/Lib/site-packages/OpenGL/DLLS"

- <http://www.cim.mcgill.ca/~fmannan/comp557/Python%20and%20PyOpenGL%20Installation.html>

- Ubuntu, MacOS

```
$ pip install PyOpenGL
```

Install Additional Modules

- GLFW
 - Windows, MacOS

```
$ pip install glfw
```

- Ubuntu

```
$ sudo apt-get install libglfw3  
$ pip install glfw
```


Install Additional Modules

- GLFW
 - Windows
 - If you are experiencing this error, download glfw library for the window at the following URL. Copy “lib-vc2015/glfw3.dll” to "python-installation-directory/Lib/site-packages/glfw"
 - <https://www.glfw.org/download.html>

```
명령 프롬프트 - py -3
Microsoft Windows [Version 10.0.17763.316]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Jaeseok>py -3
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:02:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import glfw
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Users\Jaeseok\AppData\Local\Programs\Python\Python35\Lib\site-packages\glfw\__init__.py",
line 34, in <module>
    raise ImportError("Failed to load GLFW3 shared library.")
ImportError: Failed to load GLFW3 shared library.
>>>
```

Running Python Interpreter 1

- **Interactive mode**

- Windows: Start, type “cmd”,

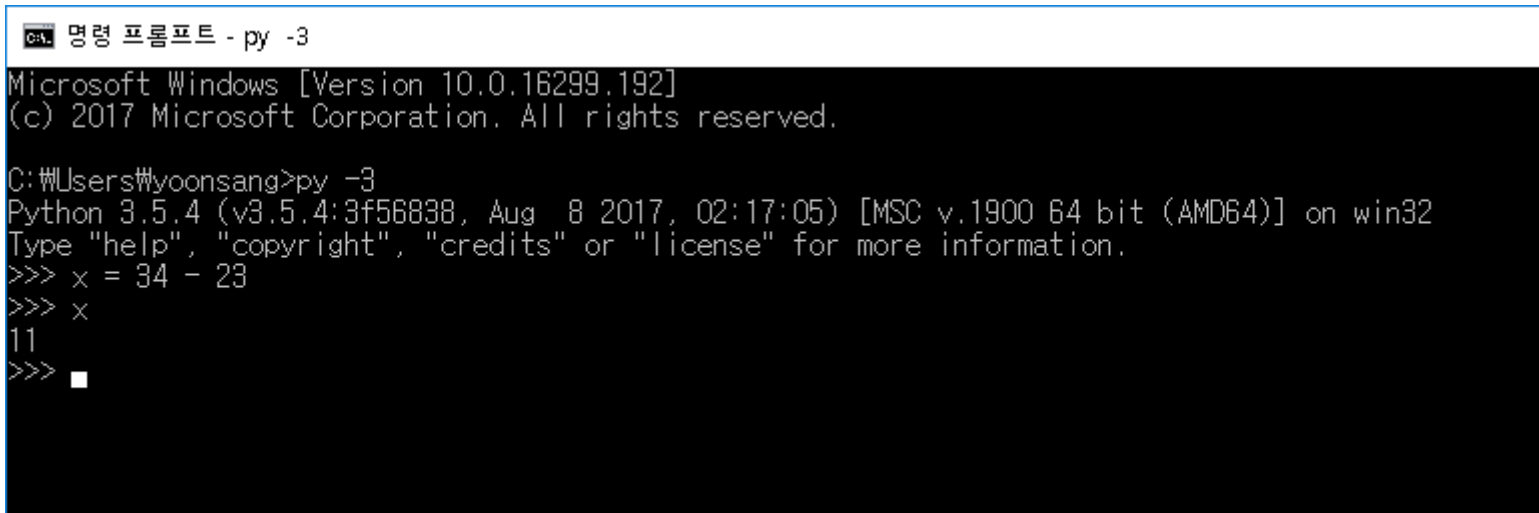
```
> py -3
```

- Ubuntu: Start, type “terminal”,

```
$ python3
```

- Suitable for simple tests

- To exit the interpreter, type `exit()` and press enter key.



```
명령 프롬프트 - py -3
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\#yoonsang>py -3
Python 3.5.4 (v3.5.4:3f56838, Aug  8 2017, 02:17:05) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 34 - 23
>>> x
11
>>> █
```

Running Python Interpreter 2

- **Non-interactive mode (runs a source file)**
 - Windows `> py -3 test.py`
 - Ubuntu `$ python3 test.py`
 - In most cases, you will use this mode.
- You can write a Python source file using your favorite editor.
 - Vim, Notepad++, Sublime Text, Atom, IDLE ...
 - I'm personally using vim & gvim.

Python References

- <https://docs.python.org/ko/3/tutorial/index.html>
- <https://docs.python.org/3/tutorial/index.html>
- <https://www.tutorialspoint.com/python3/>

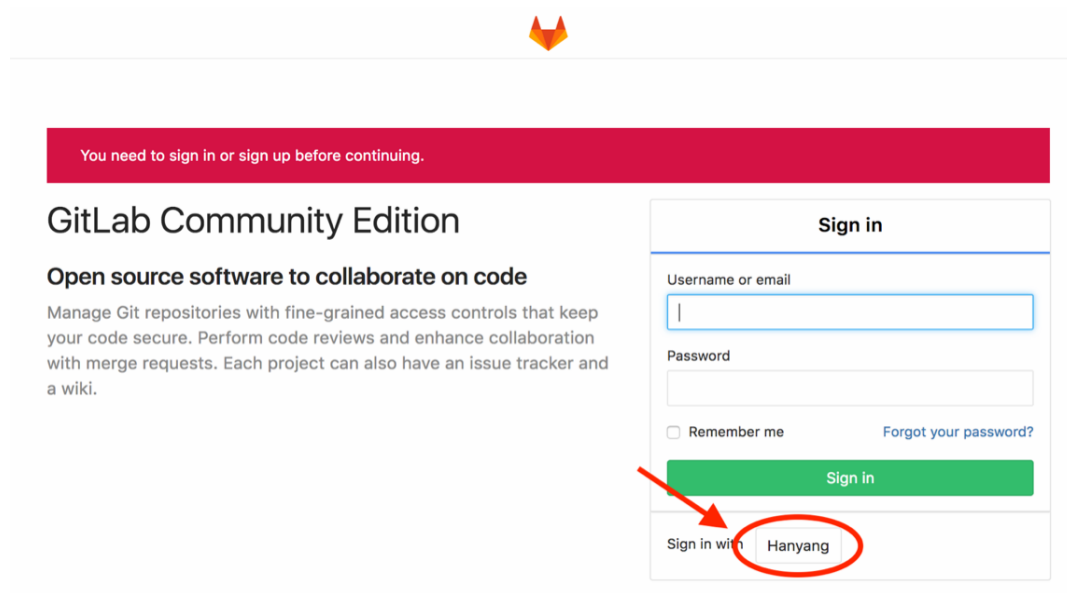
Creating a Gitlab account

Gitlab

- For today's lab assignment, submit your files via Blackboard course homepage.
- From next week's lab assignment, submit your files via the gitlab at <https://hconnect.hanyang.ac.kr/>
- Be sure to create a hconnect account in advance.
- If you already have a hconnect account, just skip this part.

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 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.

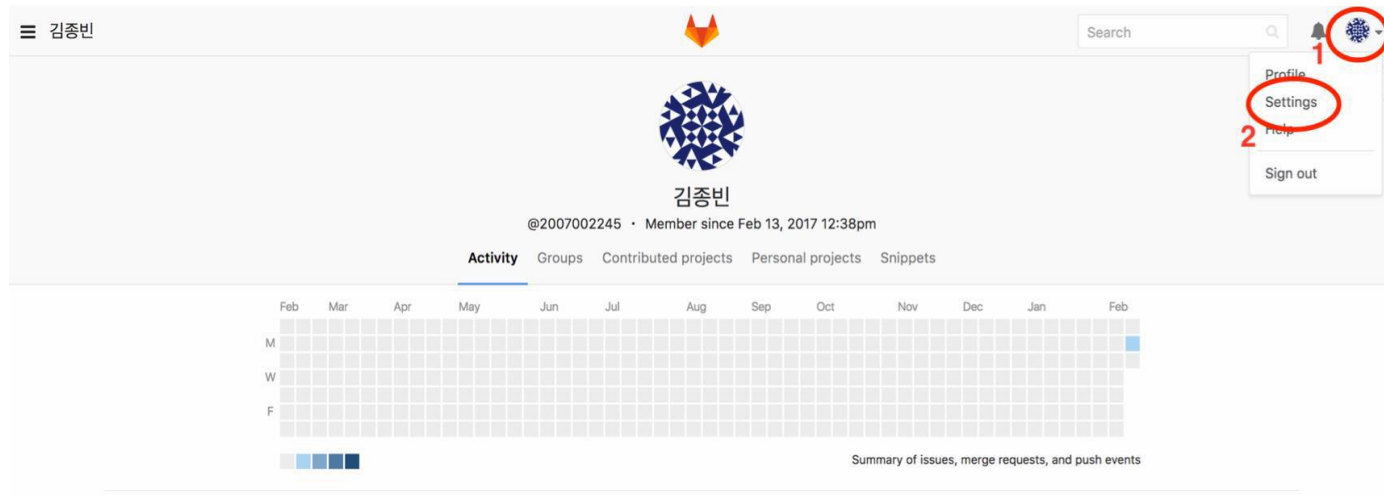
동의 취소

agree



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

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

New password

Password confirmation

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

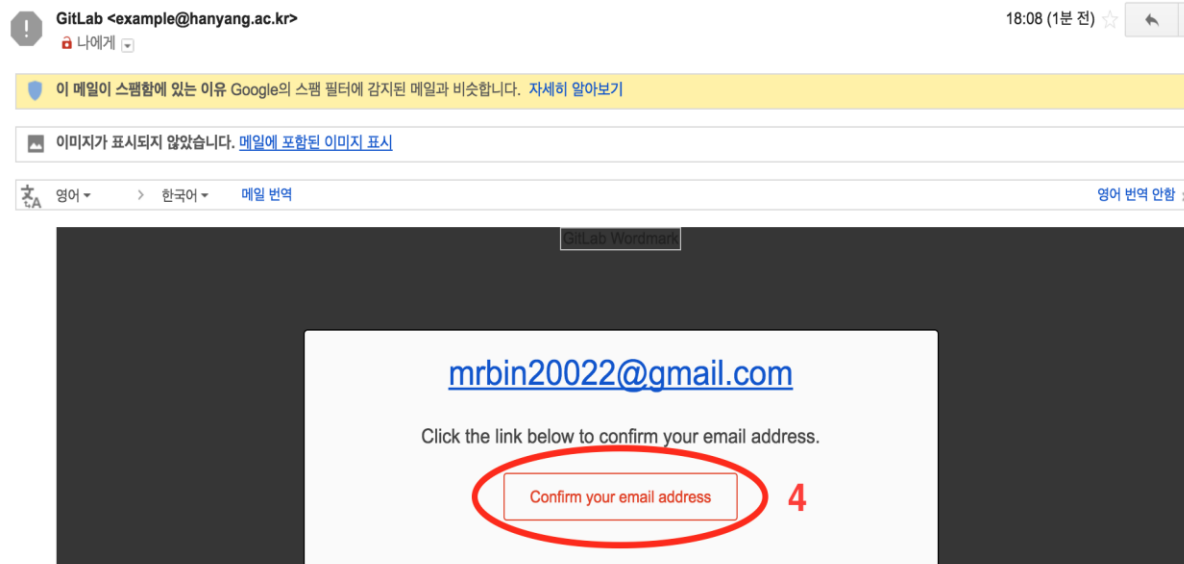
Initial password: Your HY-in portal password

- Set up Email

The screenshot shows the GitHub User Settings page. The 'Profile' tab is selected and circled in red with a '1' next to it. Below the navigation bar, there is a 'Public Avatar' section with a green and white checkered avatar. To the right is an 'Upload new avatar' section with a 'Browse file...' button and the text 'No file chosen'. Below this is the 'Main settings' section. The 'Name' field contains '김종빈'. The 'Email' field contains 'mrbin20022@gmail.com', which is circled in red with a '2' next to it. Below the email field is a 'Bio' field. At the bottom of the form, the 'Update profile settings' button is circled in red with a '3' next to it. A 'Cancel' button is also visible.

Git

- Set up Email – Approve from changed email



Git

- After this, you can sign in to hconnect with your student ID / email and the password you changed.
(without using '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

Lab Assignment 1

- Now, let's start the lab assignment 1.
- Lab assignment 1 is just for practice, will not be included in the final grade.
- However, you need to complete and submit your answers **to figure out how to set up the environment and to create your "hconnect" site account** in advance.
- Check the assignment: Blackboard course home - Lab assignments - "LabAssignment1.pdf"
- Submit your files: Blackboard course home - Lab assignments - "Lab Assignment 1, 1" and " Lab Assignment 1, 2".
- You can leave the lab after submitting your files.