

## 2. 환경설정

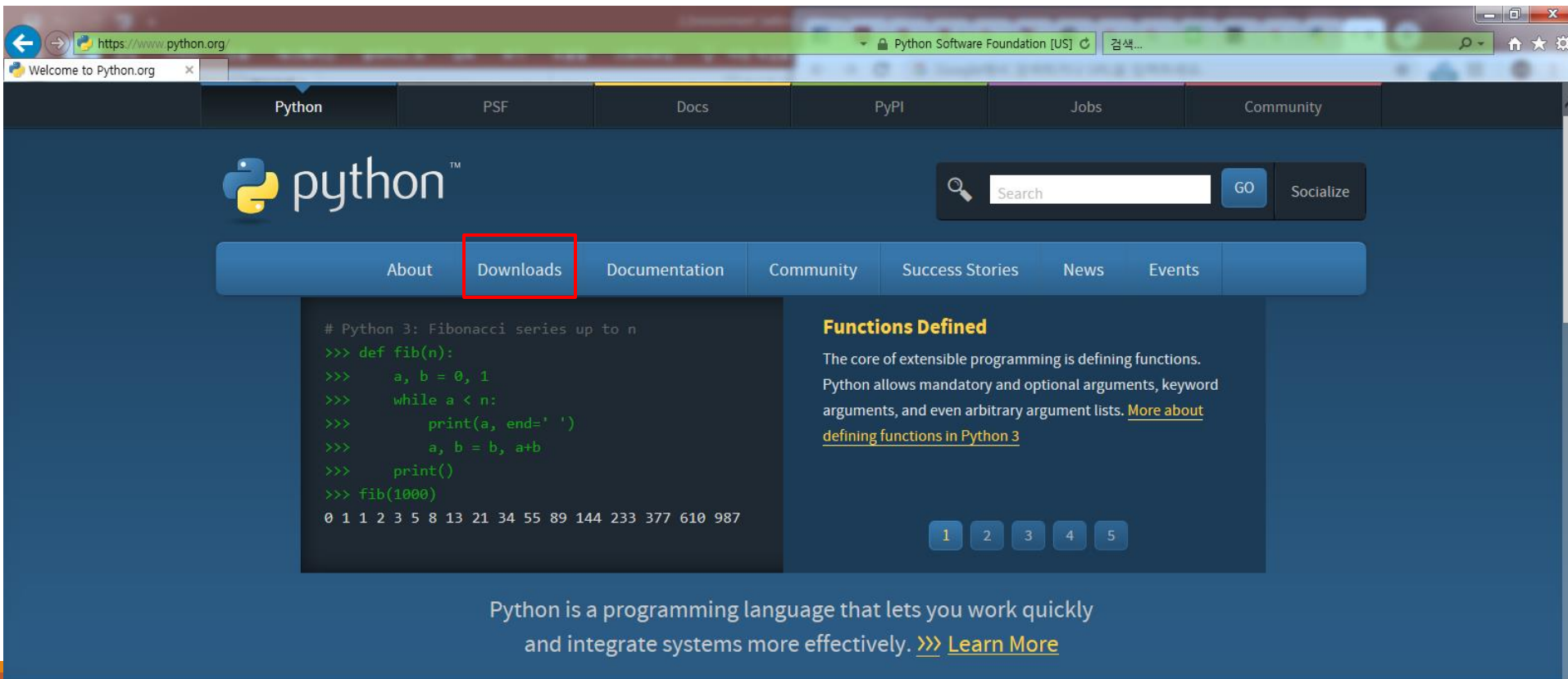
---

2018.12

일병 김재형

# Python설치(windows)

<https://www.python.org>



[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

## Download the latest version for Windows

[Download Python 3.7.1](#)

Looking for Python with a different OS? Python for [Windows](#),  
[Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)

Looking for Python 2.7? See below for specific releases



Help the sustain the Python Software Foundation!

[Participate in our Recurring Giving Campaign](#)

### Looking for a specific release?

Python releases by version number:

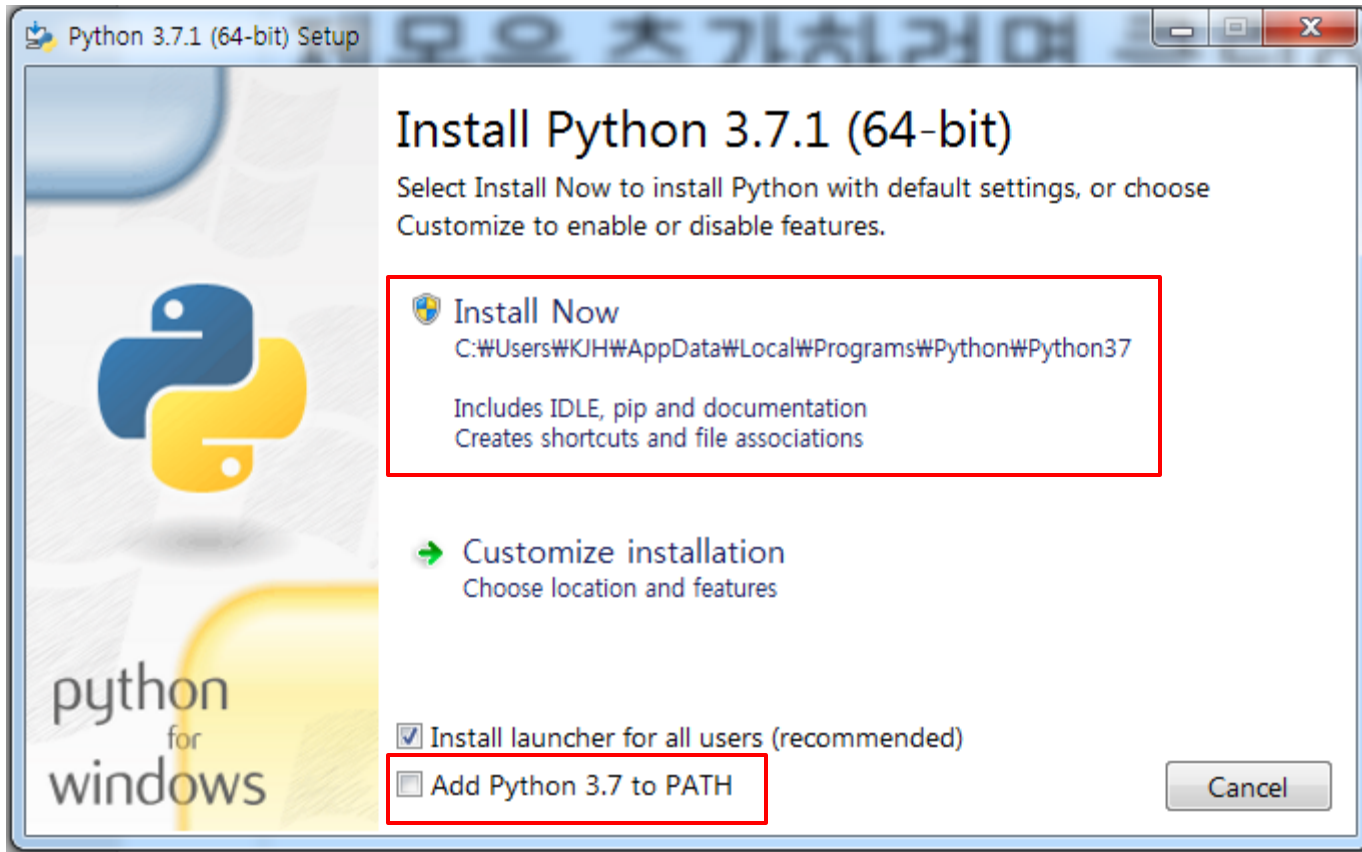
Release version	Release date		Click for more
<a href="#">Python 3.7.1</a>	2018-10-20	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.7</a>	2018-10-20	<a href="#">Download</a>	<a href="#">Release Notes</a>

# Python설치(windows)

## Files

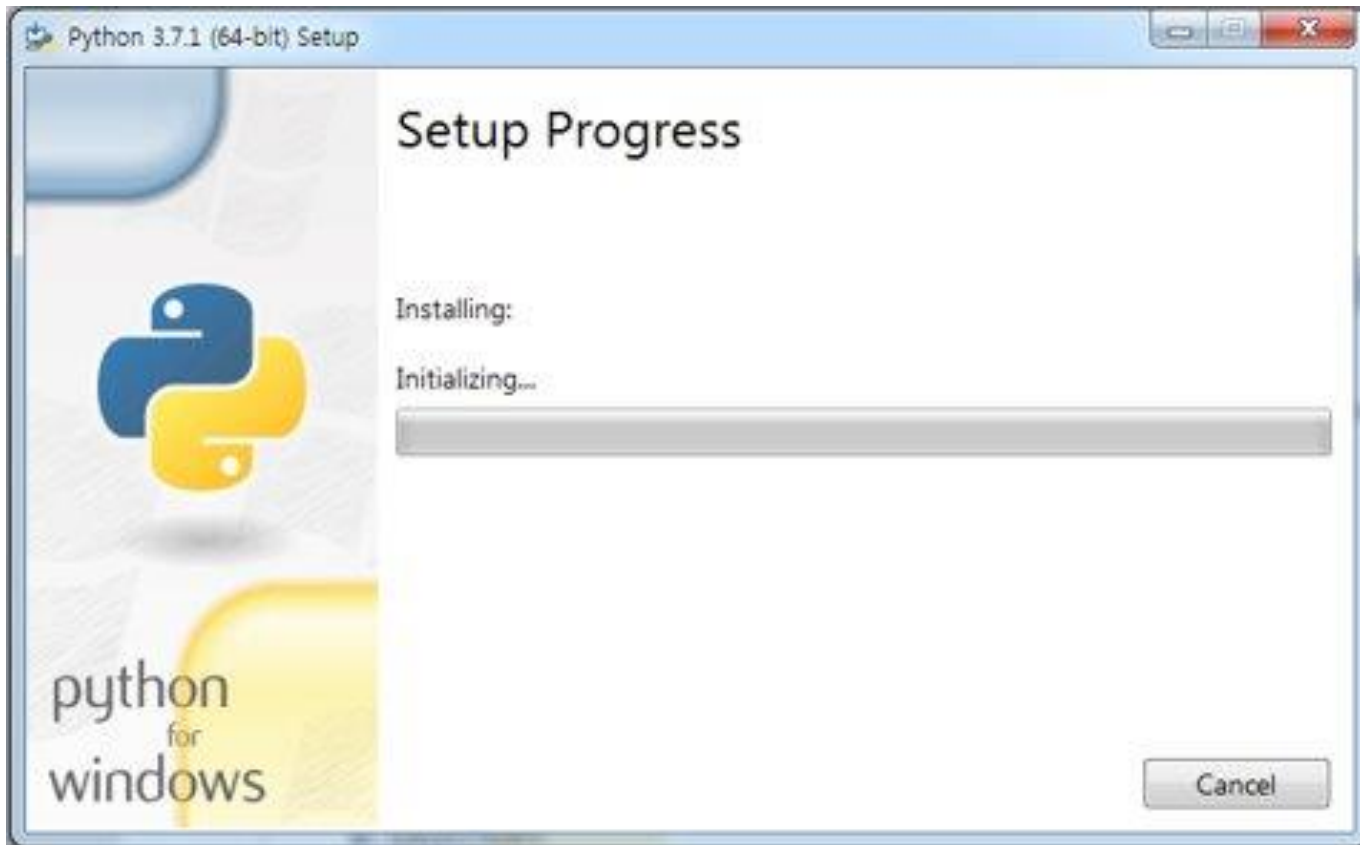
Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		99f78ecbfc766ea449c4d9e7eda19e83	22802018	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		0a57e9022c07fad3dadb2eef58568edb	16960060	<a href="#">SIG</a>
<a href="#">macOS 64-bit/32-bit installer</a>	Mac OS X	for Mac OS X 10.6 and later	ac6630338b53b9e5b9dbb1bc2390a21e	34360623	<a href="#">SIG</a>
<a href="#">macOS 64-bit installer</a>	Mac OS X	for OS X 10.9 and later	b69d52f22e73e1fe37322337eb199a53	27725111	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		b5ca69aa44aa46cdb8cf2b527d699740	8534435	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	74f919be8add2749e73d2d91eb6d1da5	6879900	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	4c9fd65b437ad393532e57f15ce832bc	26260496	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	6d866305db7e3d523ae0eb252ebd9407	1333960	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		aa4188ea480a64a3ea87e72e09f4c097	6377805	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		da24541f28e4cc133c53f0638459993c	25537464	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		20b163041935862876433708819c97db	1297224	<a href="#">SIG</a>

# Python설치(windows)



# Python설치(windows)

---



# IDE

---

## 통합 개발 환경

(Integrated Development Environment, IDE)

- 코딩, 디버그, 컴파일, 배포 등 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 소프트웨어

# IDE

Python설치시 기본 포함

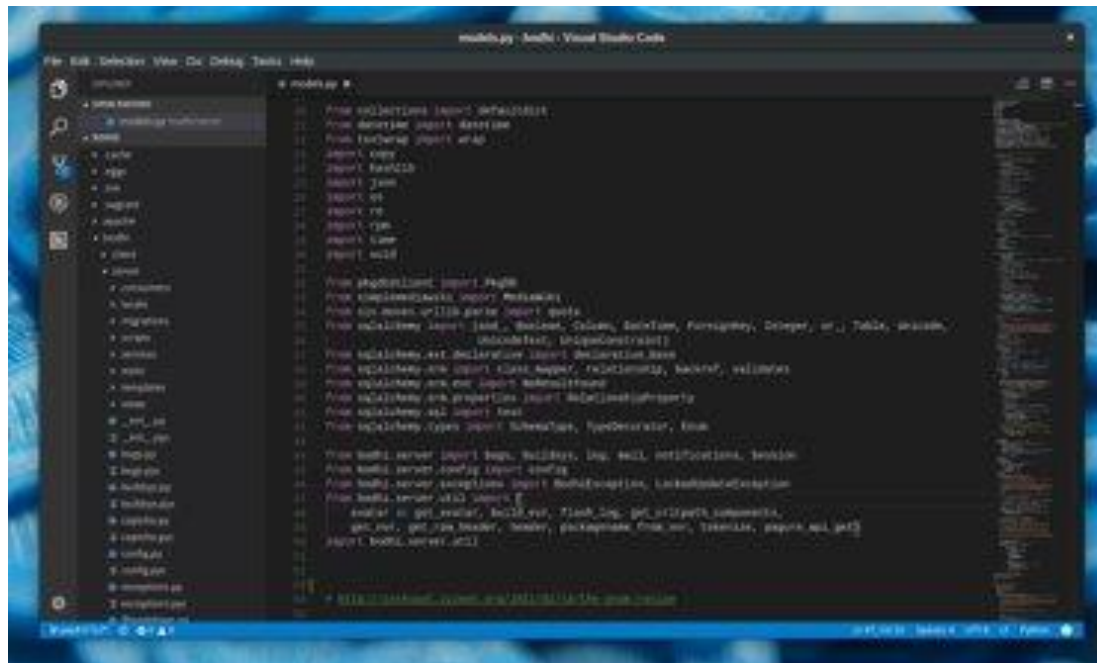
IDLE - 배터리 포함





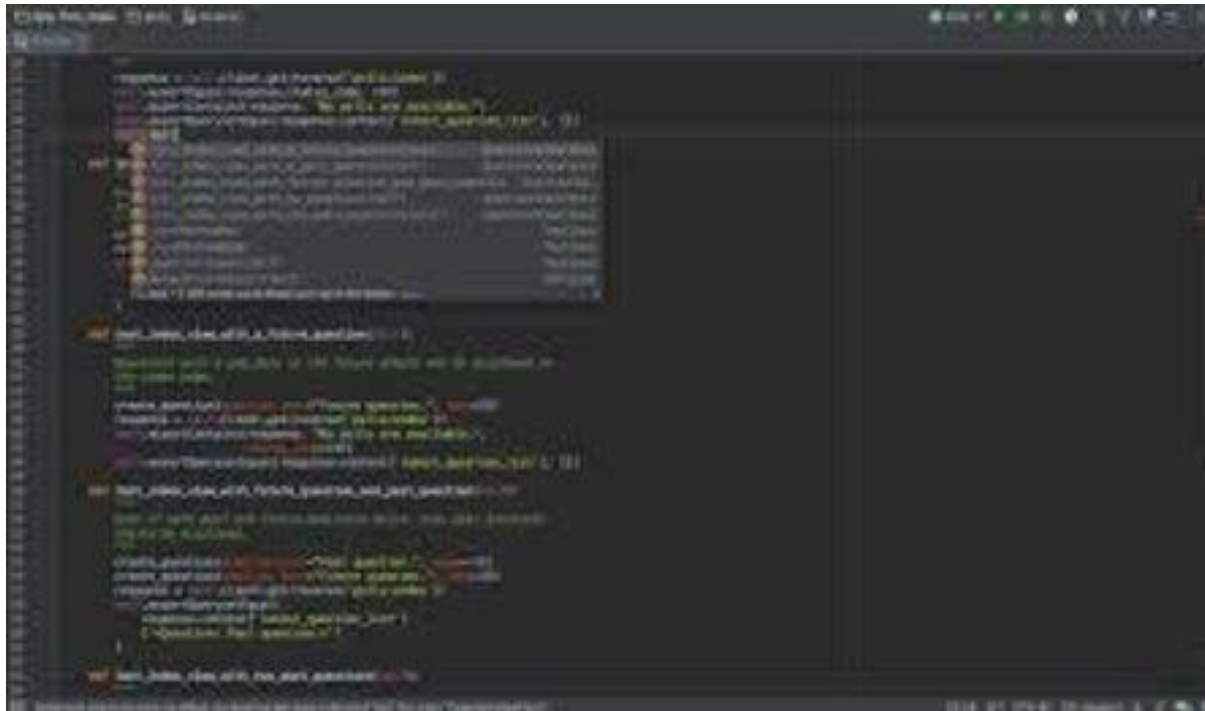
# IDE

## Visual studio



# IDE

## Pycharm



# 에디터(Editor)

---

텍스트 파일을 편집할 수 있는 프로그램.

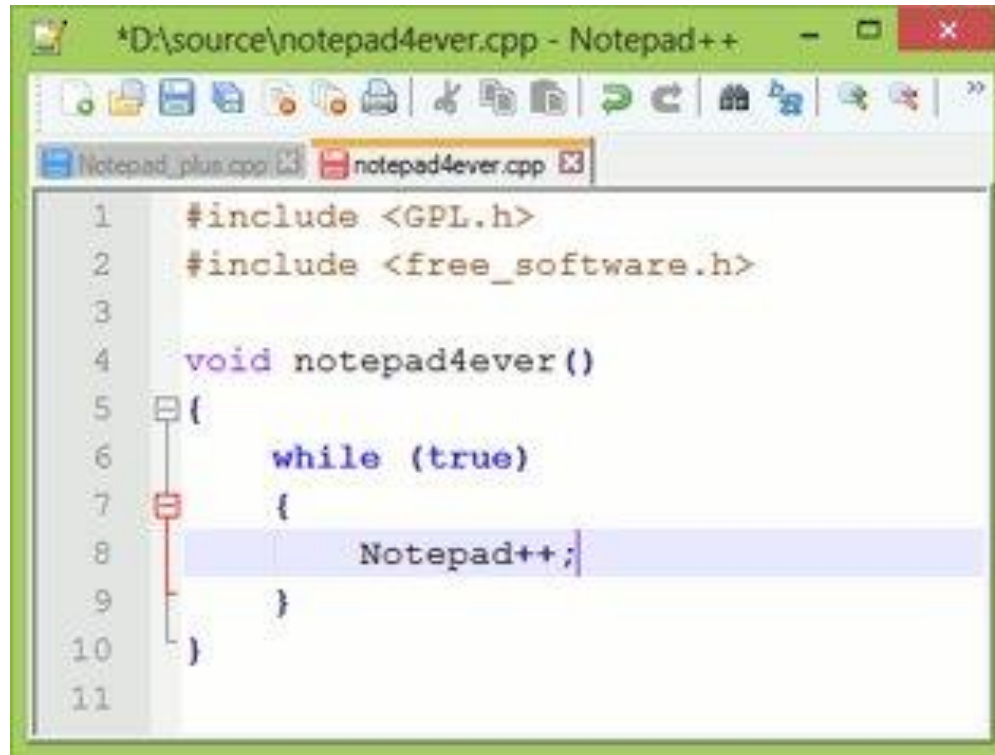
소프트웨어 개발에서 많이 쓰인다.

다만, 컴파일러나 인터프리터는 따로 설치해야 한다.

# 에디터(Editor)

---

Notepad++

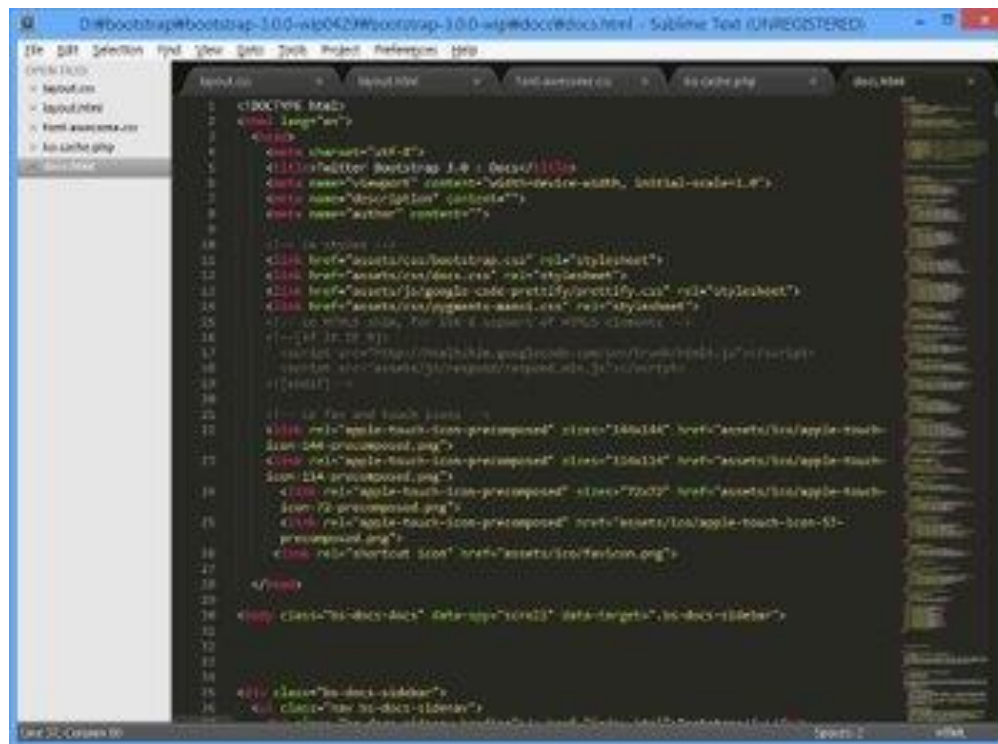
A screenshot of the Notepad++ application window. The title bar shows the file path \*D:\source\notepad4ever.cpp - Notepad++. The menu bar includes File, Edit, Format, View, and Plugins. The toolbar contains icons for opening, saving, printing, and other editing functions. The tab bar shows two open files: Notepad\_plus.cpp and notepad4ever.cpp. The main text area displays the following C++ code:

```
1  #include <GPL.h>
2  #include <free_software.h>
3
4  void notepad4ever()
5  {
6      while (true)
7      {
8          Notepad++;
9      }
10 }
11
```

The line containing 'Notepad++;' is highlighted in light blue. A vertical line on the left side of the code area indicates the current cursor position.

# 에디터(Editor)

sublimetext 3(서브라임 텍스트 3)



# 에디터(Editor)

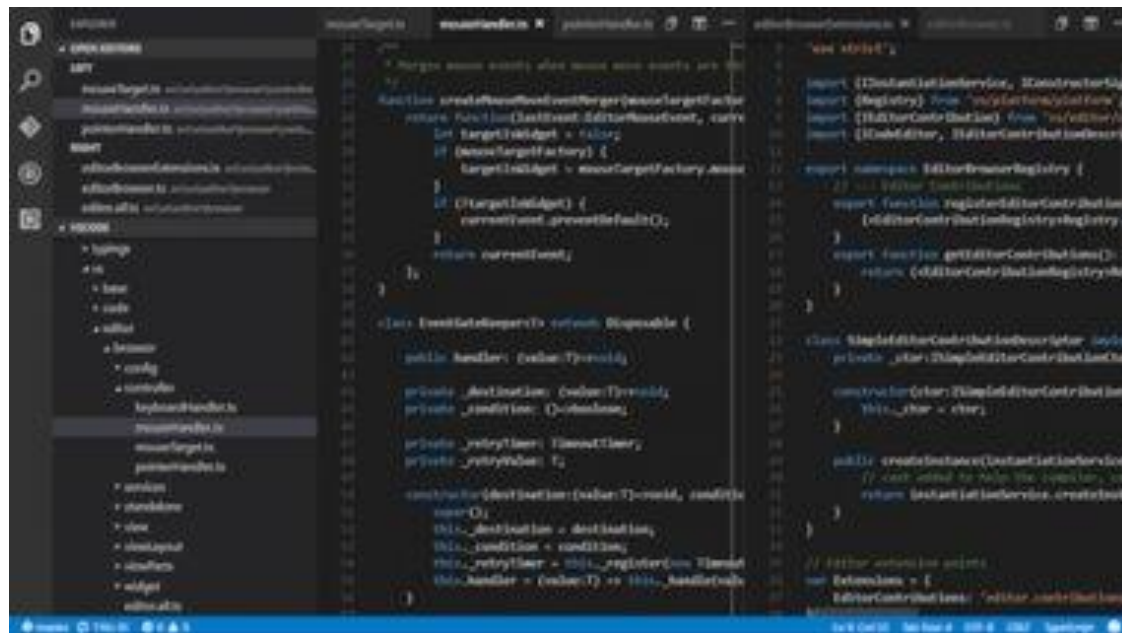
Atom(아톰)



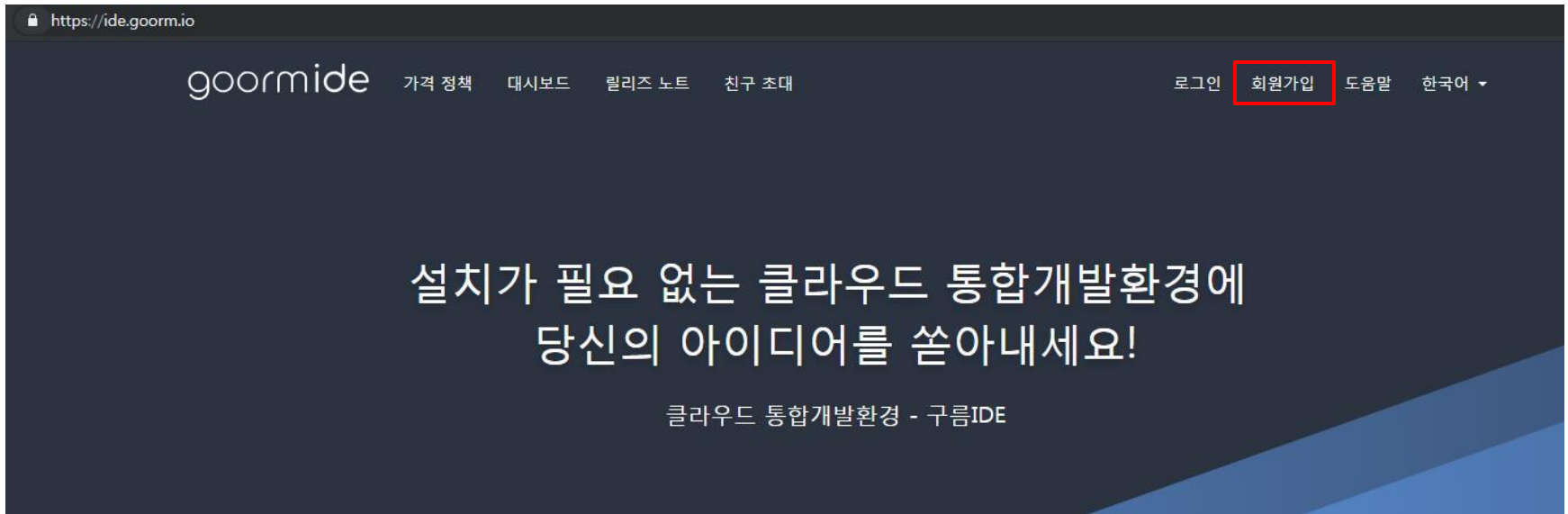
# 에디터(Editor)

비주얼 스튜디오 코드

(Visual Studio Code, VScode)



# 구름IDE 가입



구름 IDE를 검색한 후,  
<https://ide.goorm.io/>에 접속한다



# 구름IDE 가입

goorm

하나의 아이디로 구름 서비스를 이용하세요.

다른 서비스로 로그인

f G N P V

로그인

이메일

이메일 인증

비밀번호 (영문, 숫자, 특수문자 8-15자) 

비밀번호 확인 

이름 (2-15자)

☐ 구름에서 제공하는 서비스 약관에 동의합니다

간독사실

☐ 새 기능, 이벤트 정보 등의 알림 수신

이용약관의 변경이나 관계 법령에 따라 회원님께 안내되어야 할 중요 고지 사항은 메일 수신 동의 여부에 상관 없이 안내할 수 있습니다.

이미 계정이 있으신가요? 로그인

# 구름IDE 가입

goormide 가격 정책 대시보드 튜토리얼 노트 친구 초대

김재현 님 | 도움말 | 한국어

김재현

이름: 김재현

이메일: [redacted]

이메일 인증: Free | upgrade

CPU: 1코어 | 메모리: 1024MB | 저장공간: 10GB

컨테이너 생성: 0 / 1 | 워킹 커두기: 0 / 0 | 컨테이너 동시 실행: 0 / 1

**BLACK FRIDAY** goormide 프리미엄 1년 249,000원 **120,000원** 44% OFF  
11월 22일 15시 ~ 11월 30일 15시

내 컨테이너 리스트(0개)

새 컨테이너 생성

# 구름IDE 컨테이너 생성

goormide [가격 정책](#) [대시보드](#) [릴리즈 노트](#) [친구 초대](#) [김재형님](#) [도움말](#) [한국어](#)

## 컨테이너 생성

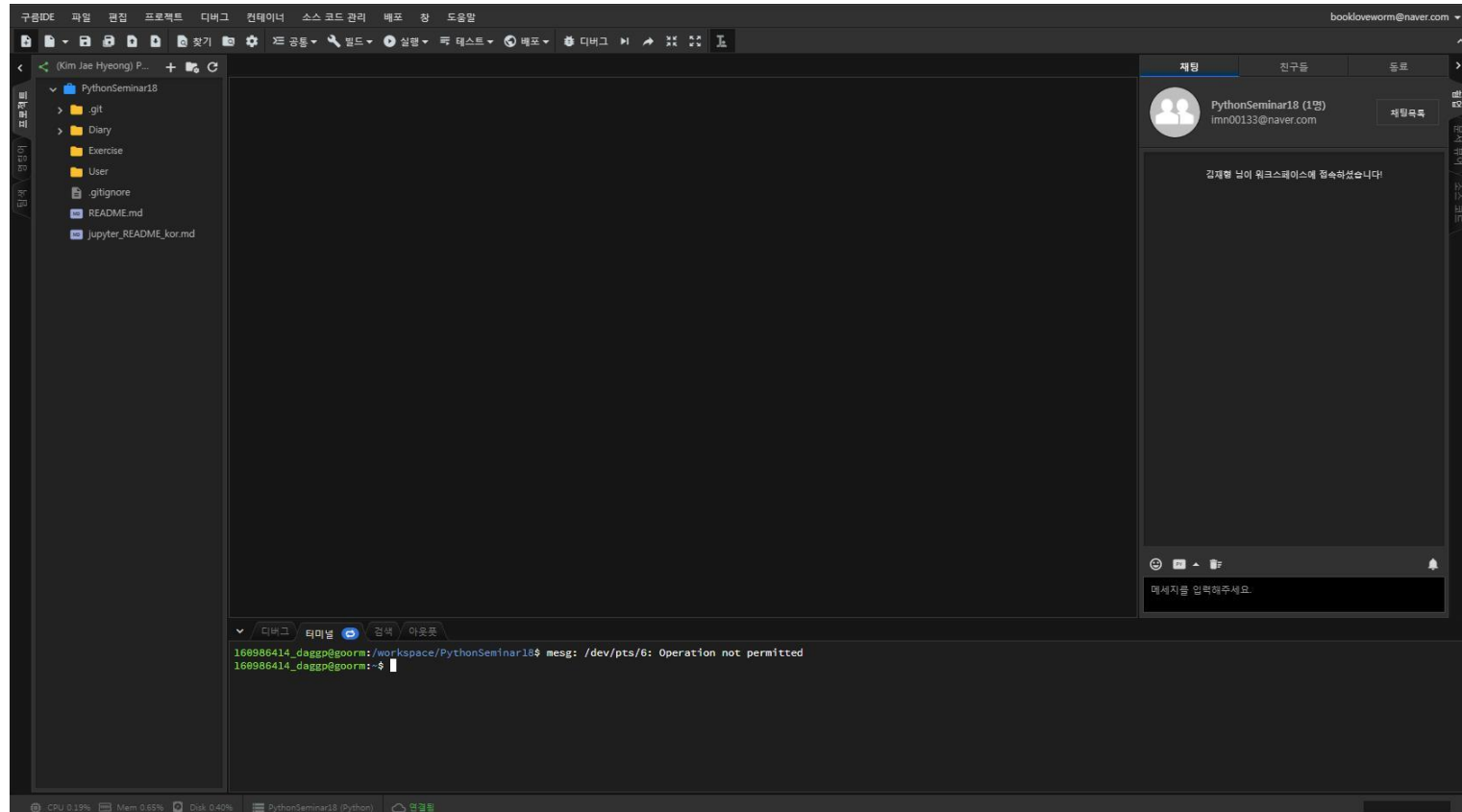
소스	<input checked="" type="radio"/> 램플릿 <input type="radio"/> Github <input type="radio"/> Bitbucket <input type="radio"/> Git / SVN <input type="radio"/> 압축파일																		
배포	<input checked="" type="radio"/> 사용 안함 <input type="radio"/> Heroku																		
이름	<input type="text" value="컨테이너 이름을 입력해주세요."/>																		
설명	<input type="text" value="컨테이너 설명을 입력해주세요."/>																		
소프트웨어 스택 선택	<div><div><input type="text" value="Search"/><ul style="list-style-type: none"><li> C/C++</li><li> Python</li><li> Django</li><li> Flask</li><li> Jupyter Notebook</li><li> TensorFlow</li><li> PyQt</li><li> JAVA</li><li> Maven</li></ul></div><div><p>Python</p><p>프로젝트 유형 <input type="text" value="Python 프로젝트"/></p><table><tr><td>OS</td><td>✓ Ubuntu 14.04 LTS</td></tr><tr><td>Python3</td><td>✓ 3.6.5</td></tr><tr><td>Python</td><td>✓ 2.7.6</td></tr><tr><td>pip3</td><td>✓ 9.0.1</td></tr><tr><td>pip</td><td>✓ 9.0.1</td></tr><tr><td>Jupyter</td><td>✓ 4.3.0</td></tr><tr><td>Django</td><td>✓ 1.11.12 LTS</td></tr><tr><td>Flask</td><td>✓ 0.12</td></tr><tr><td>TensorFlow</td><td>✓ 1.3.0</td></tr></table></div></div>	OS	✓ Ubuntu 14.04 LTS	Python3	✓ 3.6.5	Python	✓ 2.7.6	pip3	✓ 9.0.1	pip	✓ 9.0.1	Jupyter	✓ 4.3.0	Django	✓ 1.11.12 LTS	Flask	✓ 0.12	TensorFlow	✓ 1.3.0
OS	✓ Ubuntu 14.04 LTS																		
Python3	✓ 3.6.5																		
Python	✓ 2.7.6																		
pip3	✓ 9.0.1																		
pip	✓ 9.0.1																		
Jupyter	✓ 4.3.0																		
Django	✓ 1.11.12 LTS																		
Flask	✓ 0.12																		
TensorFlow	✓ 1.3.0																		

# 구름IDE 컨테이너 공유

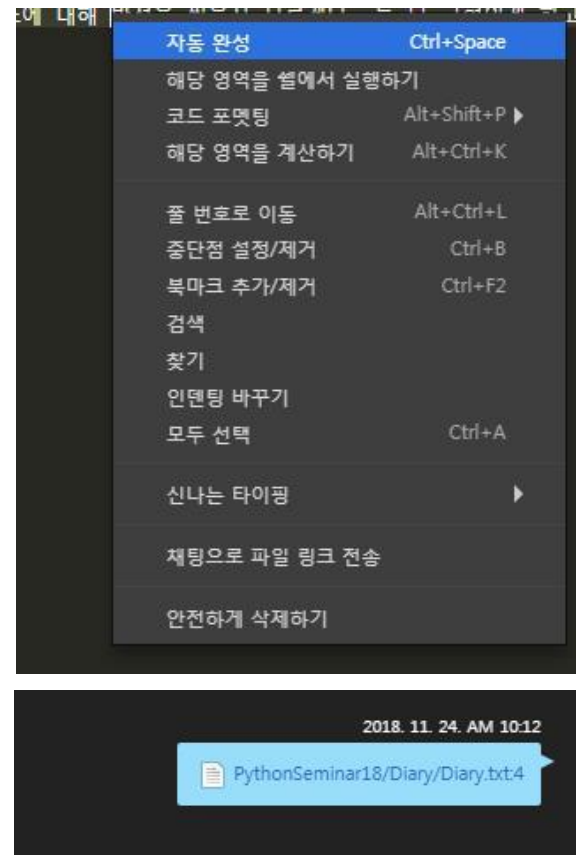
공유받은 컨테이너 리스트(1개)

PythonSeminar18	+ 더보기
 Kim Jae Hyeong님의 컨테이너 / 수정 가능 (rw 권한)	
공군에서 진행한 python수업을 위한 프로젝트. 2018.12.22 시작	
소프트웨어 스택	Python 프로젝트
저장 공간	10GB
마지막 실행 시간	2018. 11. 24. 오전 9:29:57
SSH	컨테이너 실행 시 사용 가능
테마	<input type="radio"/> Light <input checked="" type="radio"/> Dark
<div>실행</div> <div>터미널 실행</div>	

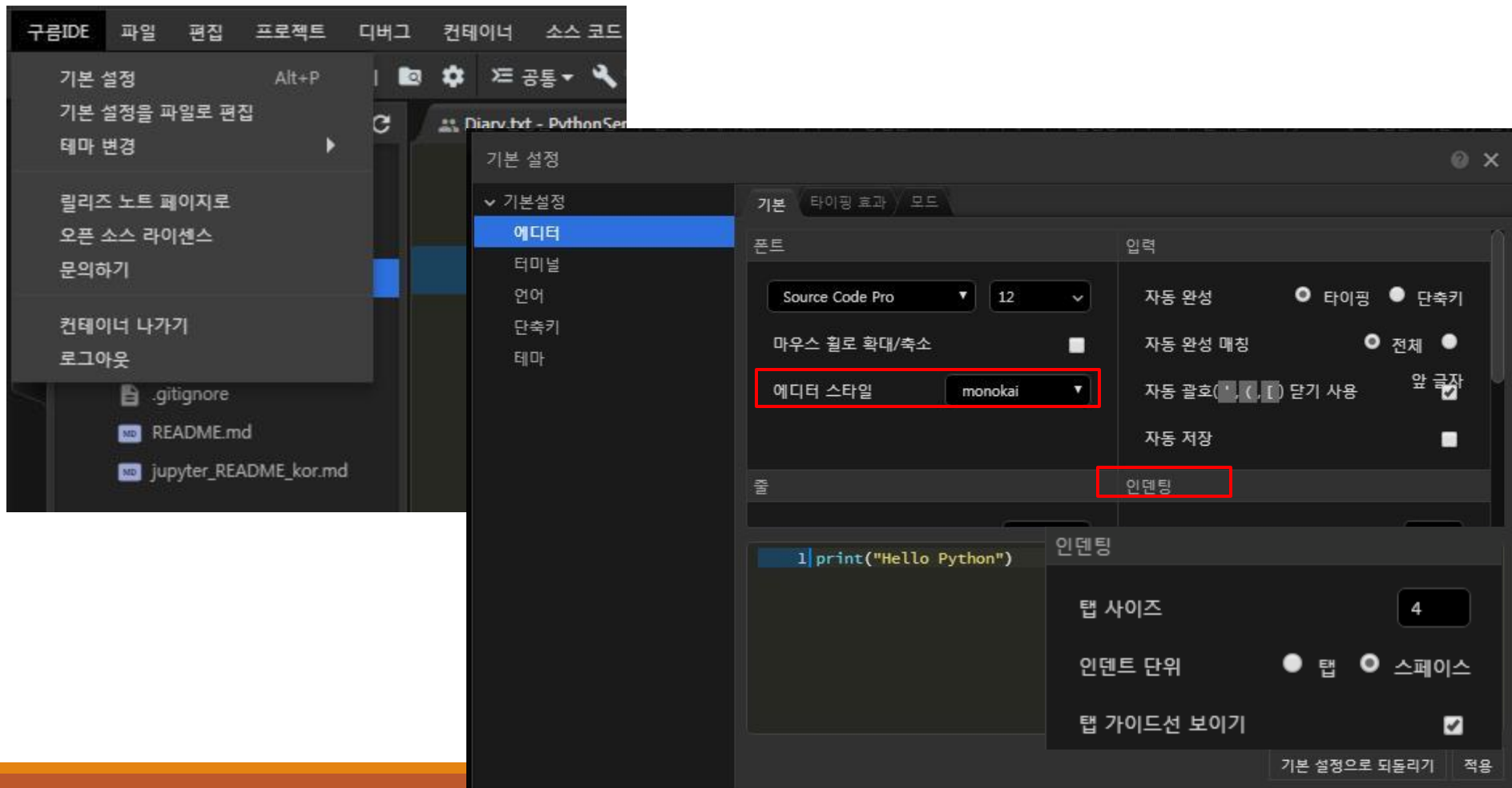
# 구름IDE 컨테이너 접속



# 구름IDE 채팅



# 구름IDE-환경설정



# 공유된 컨테이너 파일

## 강의 과제 파일 및 설명

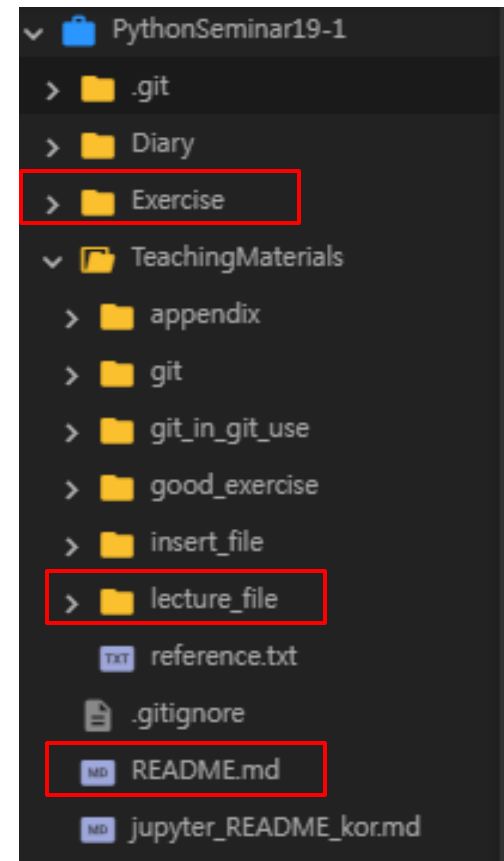
- Exercise 폴더 내 과제 이름

## 강의 파일 위치

- TeachingMaterials내 lecture\_file

## 본 세미나에 대한 설명 파일

- README.md





# 컨테이너 주의사항

---

본 컨테이너는 공개된 github를 통해 백업되고 있습니다.

- 따라서 개인정보를 본 컨테이너에 올릴 경우, 전체 공개가 되니 주의하시길 바랍니다.
- 백업은 구름 IDE에 접속을 끝내기 전에 한 번씩 진행됩니다.
- Git과 관련된 사항은 추후 진행할 예정입니다.

# Ubuntu?

---



Linux 커널

- 리눅스 배포판 중 하나

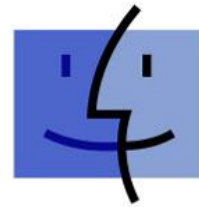
자유소프트웨어

- 배포판을 수정하거나 수정한 것을 재배포 할 수 있다.

# 운영체제

---

하드웨어와 컴퓨터 사용자 간의 매개체 역할을 하는 시스템 소프트웨어



Mac OS



# 커널과 셸

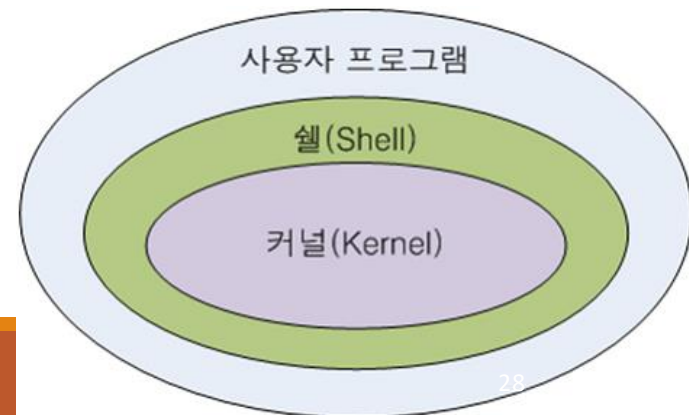
---

## 커널(Kernel)

- 운영체제(operating system)의 핵심 부분
- 시스템을 제어함

## 셸(Shell)

- 커널과 사용자간의 다리 역할
- 사용자에게서 명령을 받아 해석하여 커널에 전달

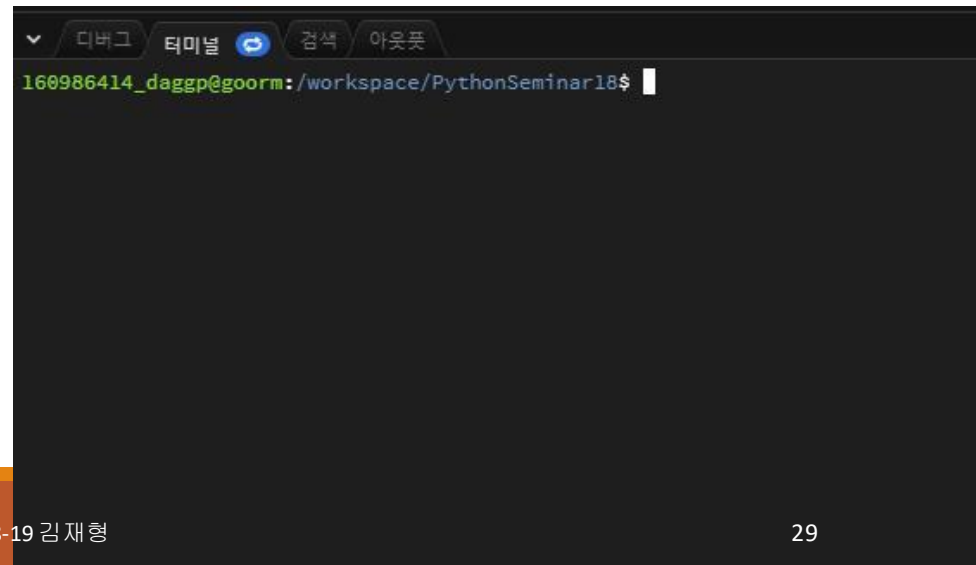


# Linux 터미널의 이해

---

## 접속 시 터미널 존재

- bash shell로 연결
- 녹색: 사용자 이름
- 파랑색: 현재 위치
- \$: 현재 shell의 권한(일반)



# CLI & GUI

---

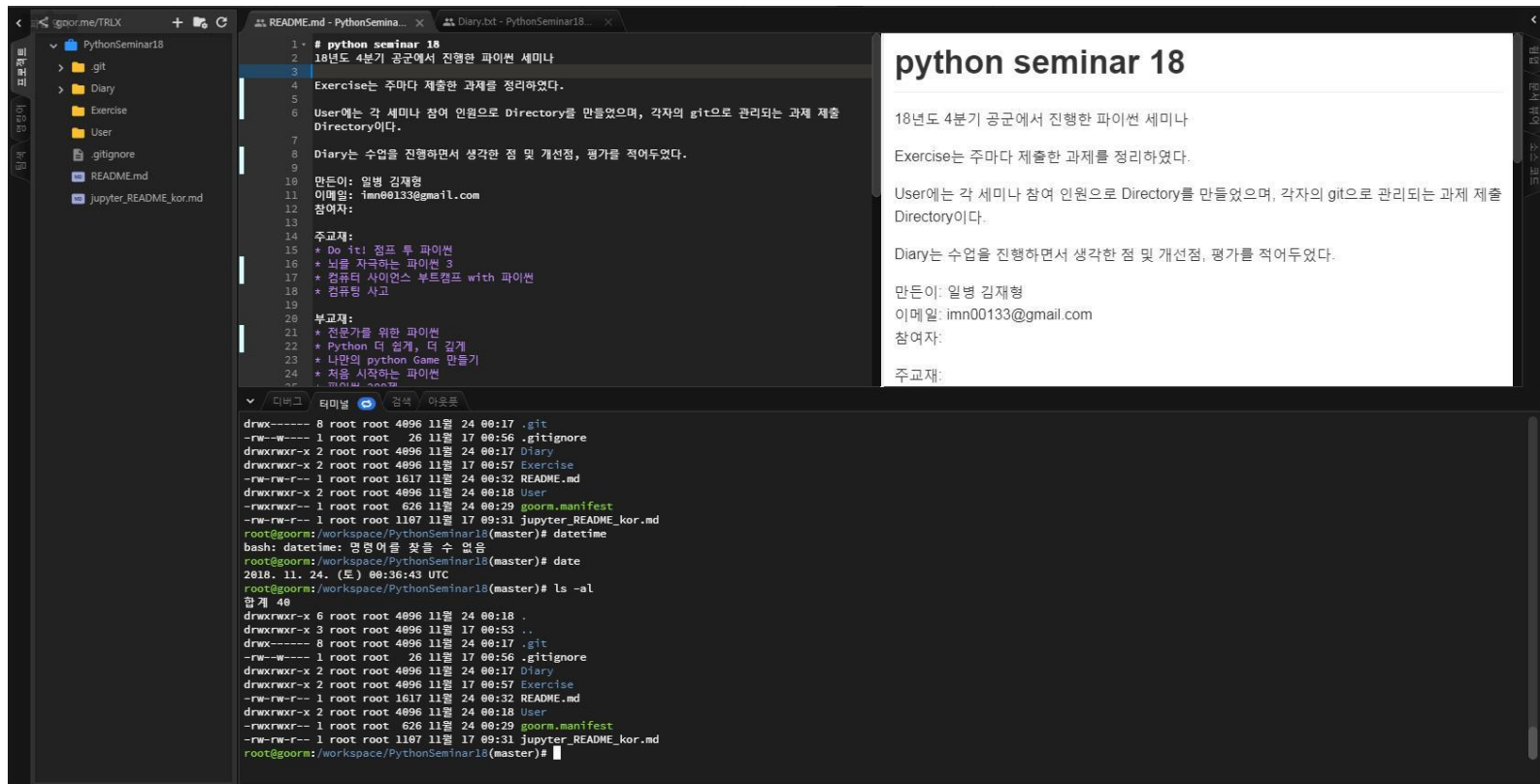
## CLI(Command line interface)

- 작업 명령을 키보드 등을 통해 문자열의 형태로 입력하고, 작업에 대한 내용도 문자열의 형태로 출력된다.

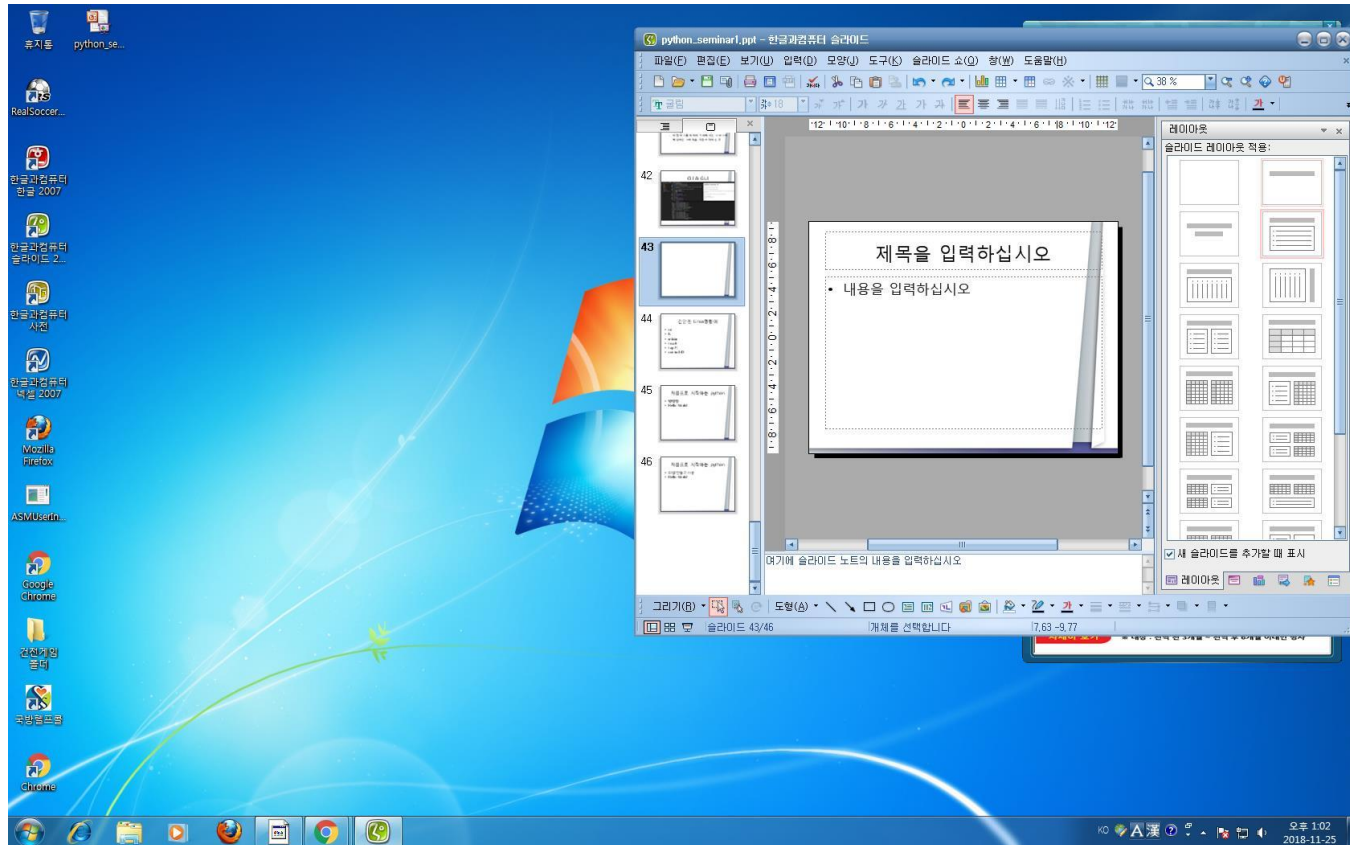
## GUI(Graphical user interface)

- 어떤 요소를 제어하기 위해서는 그 요소에 해당하는 그래픽을 직접 조작하는 것

# CLI & GUI



# CLI & GUI





# 간단한 Linux명령어

---

## 주의

- 구름IDE 컨테이너에 각 사용자별로 권한부여를 해 관리를 하려고 하였으나, 구름IDE자체가 커스텀되어있어 권한부여가 되지 않습니다.
- 자신의 과제 제출 디렉터리 외 다른 디렉터리에서 아무거나 삭제하거나 변경하지 말아주세요. 잘못하면 컨테이너 자체가 날아갑니다.

# 간단한 Linux명령어

---

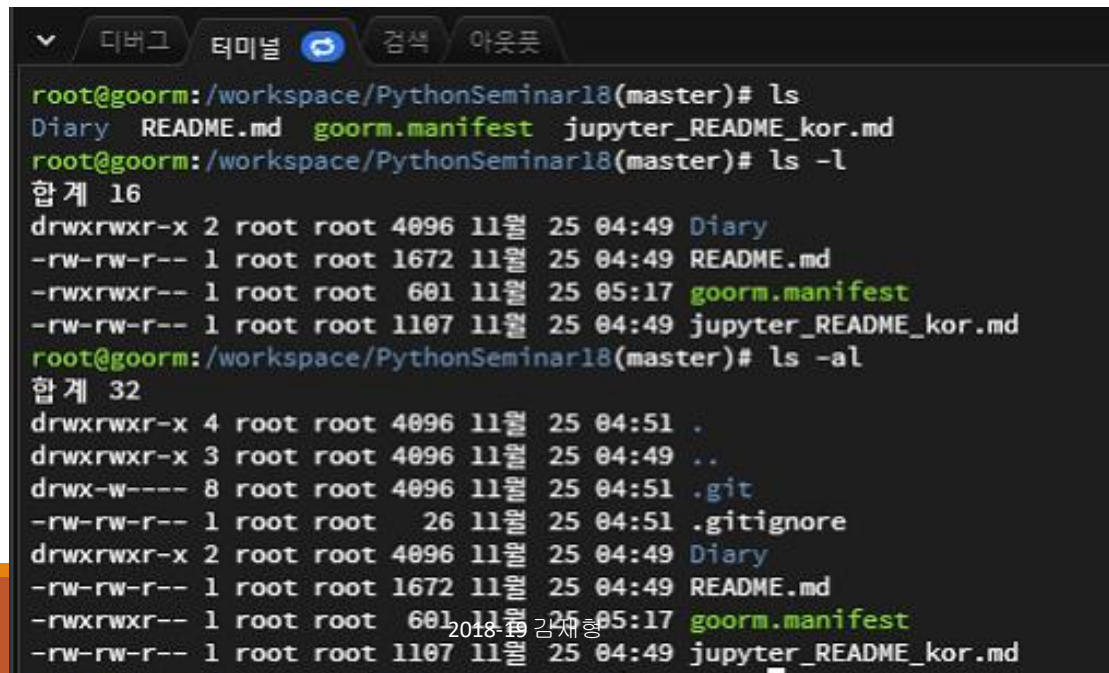
- ls
- cd
- mkdir
- touch
- tap 키
- control+D

※ Linux에서는 파일의 대소문자를 구별합니다.

# 간단한 Linux명령어 - ls

## ls(list)

- 디렉터리 내 파일과 폴더를 확인하는 명령
- a: 숨김파일 표시(.으로 시작)
- l: 파일이나 디렉터리 정보를 길게 보여줌



```
root@goorm:/workspace/PythonSeminar18(master)# ls
Diary README.md goorm.manifest jupyter_README_kor.md
root@goorm:/workspace/PythonSeminar18(master)# ls -l
합계 16
drwxrwxr-x 2 root root 4096 11월 25 04:49 Diary
-rw-rw-r-- 1 root root 1672 11월 25 04:49 README.md
-rwxrwxr-- 1 root root 601 11월 25 05:17 goorm.manifest
-rw-rw-r-- 1 root root 1107 11월 25 04:49 jupyter_README_kor.md
root@goorm:/workspace/PythonSeminar18(master)# ls -al
합계 32
drwxrwxr-x 4 root root 4096 11월 25 04:51 .
drwxrwxr-x 3 root root 4096 11월 25 04:49 ..
drwx-w---- 8 root root 4096 11월 25 04:51 .git
-rw-rw-r-- 1 root root 26 11월 25 04:51 .gitignore
drwxrwxr-x 2 root root 4096 11월 25 04:49 Diary
-rw-rw-r-- 1 root root 1672 11월 25 04:49 README.md
-rwxrwxr-- 1 root root 601 11월 25 05:17 goorm.manifest
-rw-rw-r-- 1 root root 1107 11월 25 04:49 jupyter_README_kor.md
```

# 간단한 Linux명령어 -cd

---

cd(change directory)

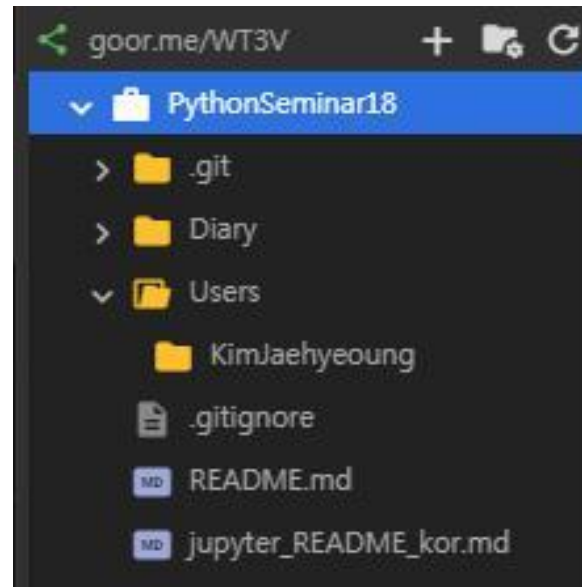
- 디렉터리를 이동하는 명령
- 작업중인 디렉터리 변경시 사용
- 절대 경로: 최상위 디렉터리 '/'부터 전체 경로 지정
- 상대 경로: 현재 디렉터리를 기준으로 이동
  - '.': 현재 디렉터리
  - '..': 상위 디렉터리
  - '~': 홈디렉터리(구름 IDE에서는 사용하지 않음)

# 간단한 Linux명령어 -cd

---

```
root@goorm:/workspace/PythonSeminar18(master)# ls -al
합계 32
drwxrwxr-x 4 root root 4096 11월 25 04:51 .
drwxrwxr-x 3 root root 4096 11월 25 04:49 ..
drwx-w---- 8 root root 4096 11월 25 09:41 .git
-rw-rw-r-- 1 root root 26 11월 25 04:51 .gitignore
drwxrwxr-x 2 root root 4096 11월 25 04:49 Diary
-rw-rw-r-- 1 root root 1672 11월 25 04:49 README.md
-rwxrwxr-- 1 root root 601 11월 25 09:37 goorm.manifest
-rw-rw-r-- 1 root root 1107 11월 25 04:49 jupyter_README_kor.md
root@goorm:/workspace/PythonSeminar18(master)# cd Diary
root@goorm:/workspace/PythonSeminar18/Diary(master)# ls -al
합계 12
drwxrwxr-x 2 root root 4096 11월 25 04:49 .
drwxrwxr-x 4 root root 4096 11월 25 04:51 ..
-rw-rw-r-- 1 root root 692 11월 25 04:49 Diary.txt
root@goorm:/workspace/PythonSeminar18/Diary(master)# cd ..
root@goorm:/workspace/PythonSeminar18(master)#
```

# 간단한 Linux명령어 -cd



```
root@goorm: /workspace/PythonSeminar18(master) # cd Users/KimJaehyeoung/  
root@goorm: /workspace/PythonSeminar18/Users/KimJaehyeoung(master) #
```

# 간단한 Linux명령어 - mkdir

---

mkdir [option] directory\_name(s)

- 디렉터리를 생성하는 명령
- 여러 이름을 입력하면 한 번에 여러 디렉터리를 생성할 수 있다.

# 간단한 Linux명령어 - mkdir

```
root@goorm:/workspace/PythonSeminar18(master)# ls -al
합계 32
drwxrwxr-x 4 root root 4096 11월 25 04:51 .
drwxrwxr-x 3 root root 4096 11월 25 04:49 ..
drwx-w---- 8 root root 4096 11월 25 09:41 .git
-rw-rw-r-- 1 root root 26 11월 25 04:51 .gitignore
drwxrwxr-x 2 root root 4096 11월 25 04:49 Diary
-rw-rw-r-- 1 root root 1672 11월 25 04:49 README.md
-rwxrwxr-- 1 root root 601 11월 25 09:37 goorm.manifest
-rw-rw-r-- 1 root root 1107 11월 25 04:49 jupyter_README_kor.md
root@goorm:/workspace/PythonSeminar18(master)# mkdir Users
root@goorm:/workspace/PythonSeminar18(master)# ls -al
합계 36
drwxrwxr-x 5 root root 4096 11월 25 10:15 .
drwxrwxr-x 3 root root 4096 11월 25 04:49 ..
drwx-w---- 8 root root 4096 11월 25 09:41 .git
-rw-rw-r-- 1 root root 26 11월 25 04:51 .gitignore
drwxrwxr-x 2 root root 4096 11월 25 04:49 Diary
-rw-rw-r-- 1 root root 1672 11월 25 04:49 README.md
drwxrwxr-x 2 root root 4096 11월 25 10:15 Users
-rwxrwxr-- 1 root root 601 11월 25 09:37 goorm.manifest
-rw-rw-r-- 1 root root 1107 11월 25 04:49 jupyter_README_kor.md
root@goorm:/workspace/PythonSeminar18(master)#
```



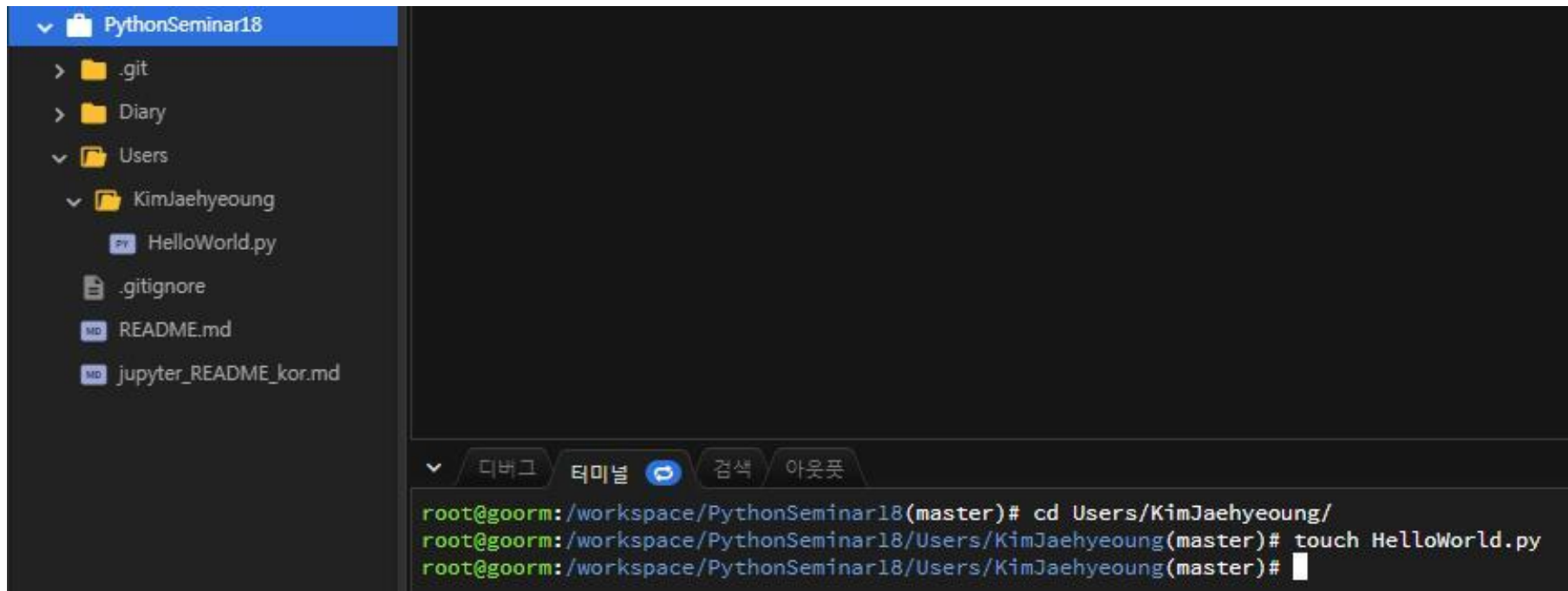
# 간단한 Linux명령어 - touch

---

touch [option] file\_name

- 파일의 최종 접근 시간, 수정시간과 같은 타임스탬프를 변경
- 파일의 크기가 0인 빈(empty)파일을 생성

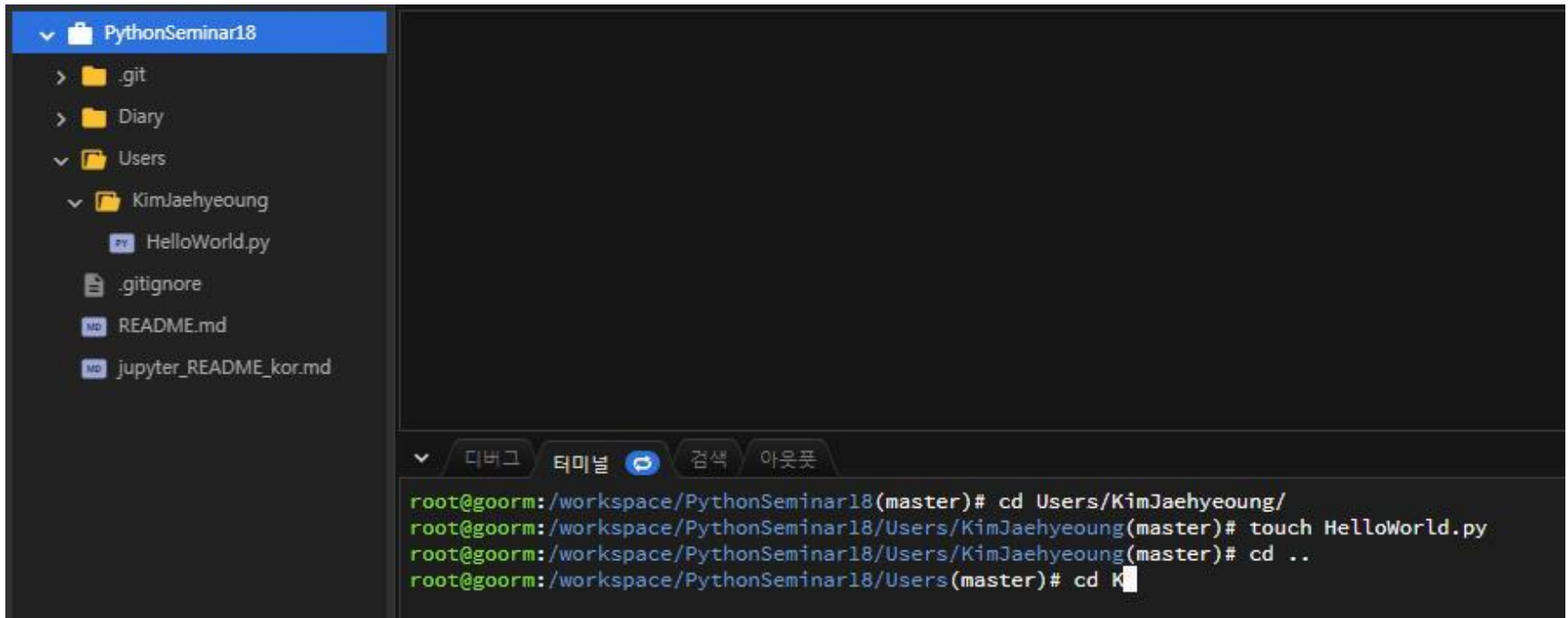
# 간단한 Linux명령어 - touch



```
root@goorm:/workspace/PythonSeminar18(master)# cd Users/KimJaehyeoung/
root@goorm:/workspace/PythonSeminar18/Users/KimJaehyeoung(master)# touch HelloWorld.py
root@goorm:/workspace/PythonSeminar18/Users/KimJaehyeoung(master)#
```

# 간단한 Linux명령어 - tap키

입력하는 도중에 tap키를 누르면 나머지 부분을 채울 수 있는 만큼 채워준다.

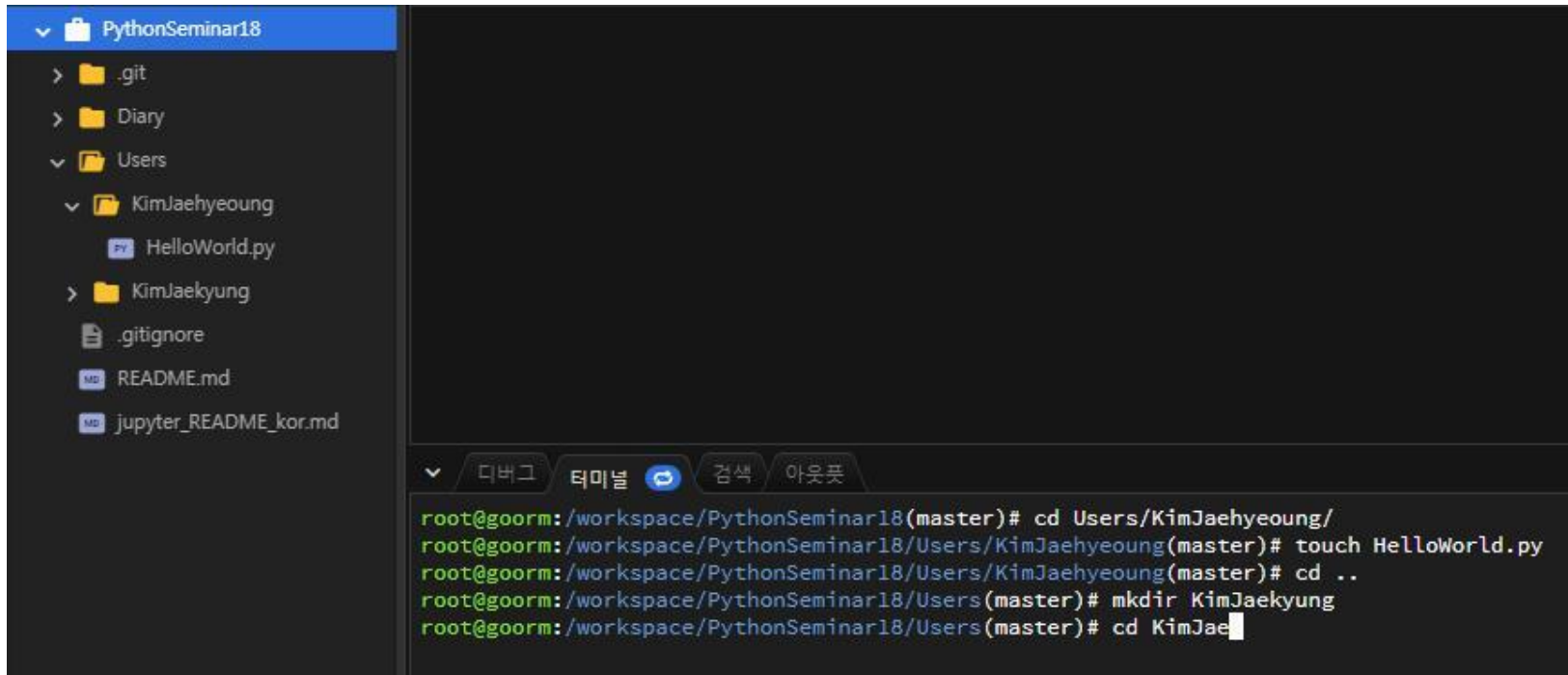


```
PythonSeminar18
├── .git
├── Diary
├── Users
├── KimJaehyeoung
│   ├── HelloWorld.py
│   ├── .gitignore
│   ├── README.md
│   └── jupyter_README_kor.md
└──
```

```
root@goorm:/workspace/PythonSeminar18(master)# cd Users/KimJaehyeoung/
root@goorm:/workspace/PythonSeminar18/Users/KimJaehyeoung(master)# touch HelloWorld.py
root@goorm:/workspace/PythonSeminar18/Users/KimJaehyeoung(master)# cd ..
root@goorm:/workspace/PythonSeminar18/Users(master)# cd K
```

# 간단한 Linux명령어 - tap키

입력하는 도중에 tap키를 누르면 나머지 부분을 채울 수 있는 만큼 채워준다.



```
root@goorm:/workspace/PythonSeminar18(master)# cd Users/KimJaehyeoung/
root@goorm:/workspace/PythonSeminar18/Users/KimJaehyeoung(master)# touch HelloWorld.py
root@goorm:/workspace/PythonSeminar18/Users/KimJaehyeoung(master)# cd ..
root@goorm:/workspace/PythonSeminar18/Users(master)# mkdir KimJaekyung
root@goorm:/workspace/PythonSeminar18/Users(master)# cd KimJae
```

# 간단한 Linux명령어 - Control+D

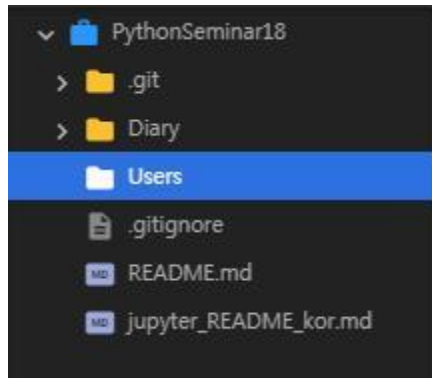
---

명령을 잘못 내렸을 때 키를 입력하면 명령을 종료한다.

- shell에서 로그아웃
- 표준 입력에서 들어오는 입력을 끝내는 시그널이기 때문에 종료된다.

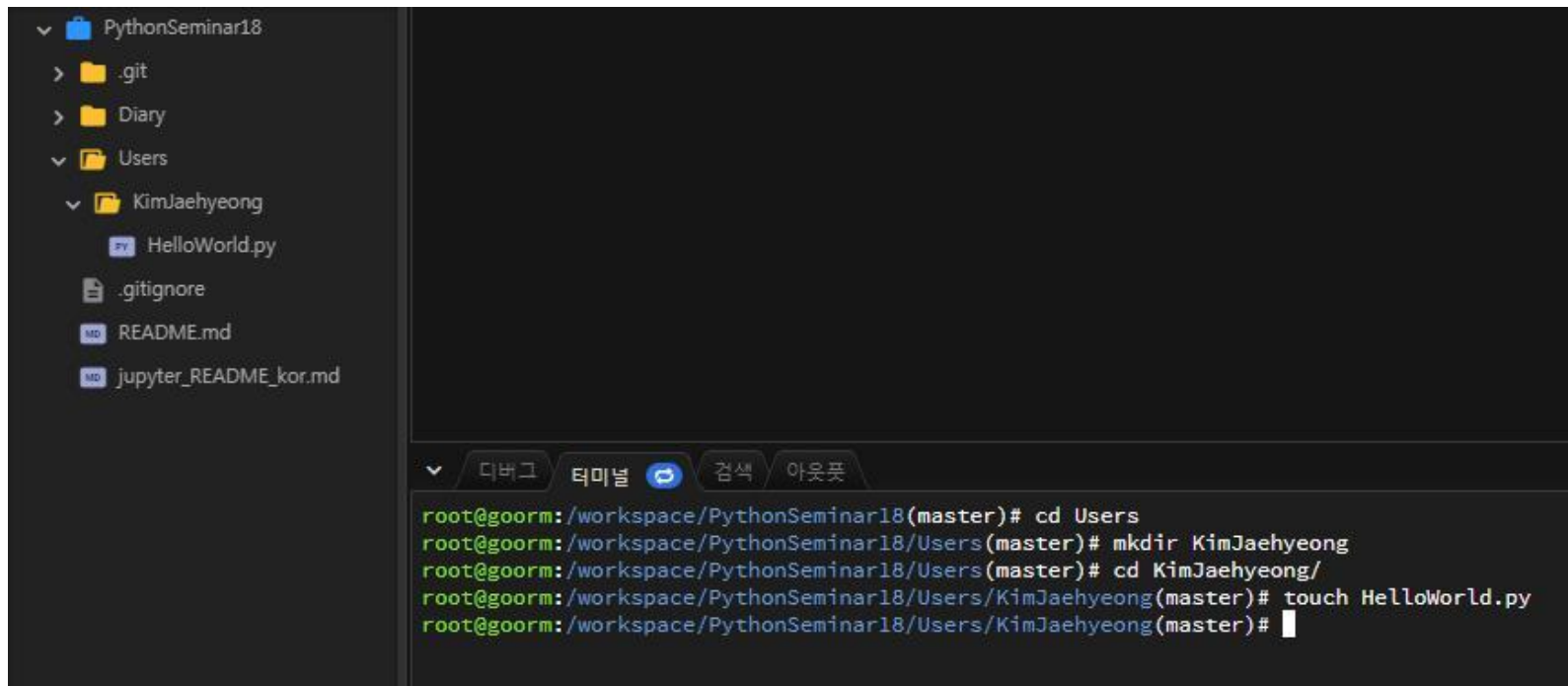
# Linux실습 - 디렉터리 만들기

---



- PythonSeminar18/Users에 자신의 영문별명으로 디렉터를 생성하시오.
- 그 후, hello\_world.py를 생성하시오.

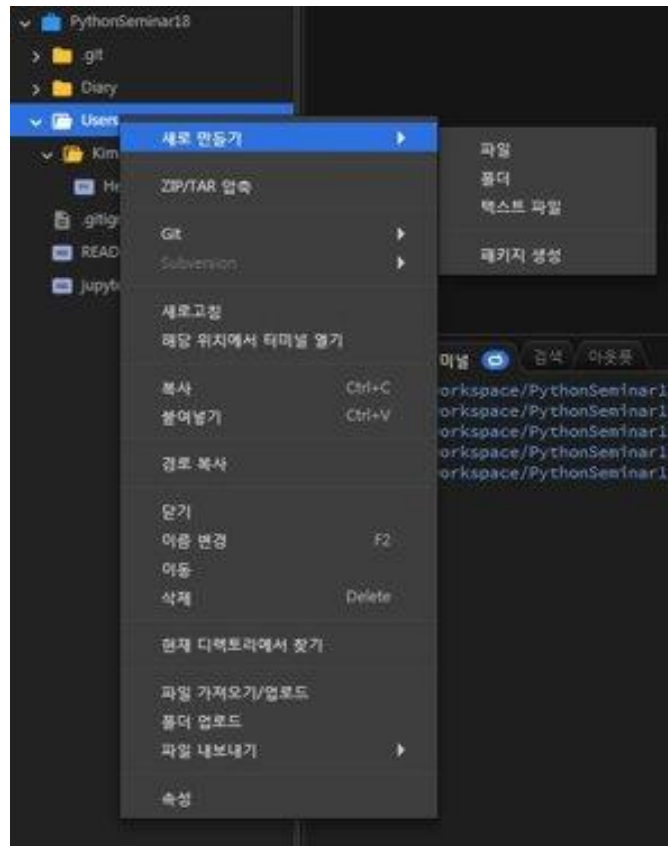
# Linux실습 - 디렉터리 만들기



The image shows a Linux environment with a file manager on the left and a terminal on the right. The file manager displays the directory structure of a project named 'PythonSeminar18', which includes subdirectories like '.git', 'Diary', 'Users', and 'KimJaehyeong'. The 'KimJaehyeong' directory contains files like 'HelloWorld.py', '.gitignore', 'README.md', and 'jupyter\_README\_kor.md'. The terminal window shows the execution of commands to create the 'KimJaehyeong' directory and the 'HelloWorld.py' file.

```
root@goorm:/workspace/PythonSeminar18(master)# cd Users
root@goorm:/workspace/PythonSeminar18/Users(master)# mkdir KimJaehyeong
root@goorm:/workspace/PythonSeminar18/Users(master)# cd KimJaehyeong/
root@goorm:/workspace/PythonSeminar18/Users/KimJaehyeong(master)# touch HelloWorld.py
root@goorm:/workspace/PythonSeminar18/Users/KimJaehyeong(master)#
```

# 구름IDE에서 파일/폴더 만들기



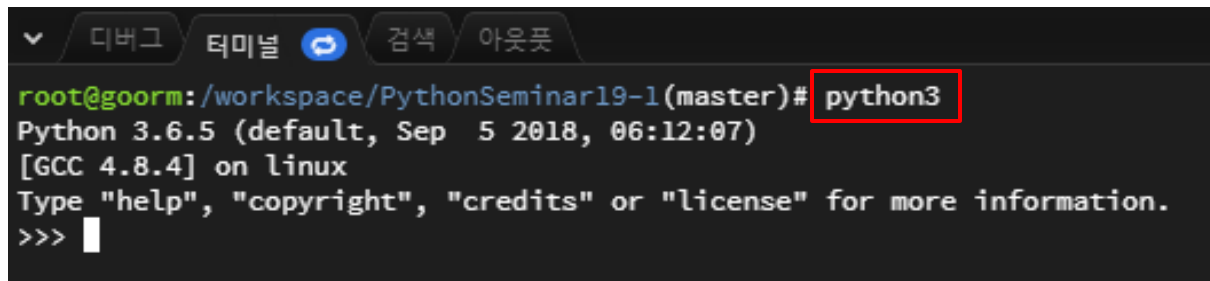


# 처음으로 시작하는 Python

---

## 구름IDE 접속

- 하단의 터미널에 python3를 입력
- 대화형 인터프리터가 나타난다.



```
▼ 디버그 터미널 🔍 아웃풋
root@goorm:/workspace/PythonSeminar19-1(master)# python3
Python 3.6.5 (default, Sep  5 2018, 06:12:07)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

# 처음으로 시작하는 Python

---

## 대화형 인터프리터

- 파이썬 셸(Python shell)이라고도 한다.
- >>>을 프롬프트(prompt)라 한다.
- 대화를 하는 것처럼 명령을 입력하면  
컴퓨터가 명령을 즉각적으로 처리해서 보여준다.

# 처음으로 시작하는 Python

---

대화형 인터프리터에서 나가기

- exit()를 입력하고 엔터를 칩니다.
- 또는 ctrl+D(unix)를 입력합니다.
- ※ 윈도우에서는 ctrl+Z

```
root@goorm:/workspace/PythonSeminar18(master)# python3
Python 3.6.5 (default, Sep  5 2018, 06:12:07)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>> exit()
root@goorm:/workspace/PythonSeminar18(master)#
```

# 처음으로 시작하는 Python

---

※ 구름IDE에는 python2와 python3가 둘 다 설치되어 python만 쳤을 때, python2가 나타날 수 있습니다.

이 경우에는 exit() 또는 ctrl+D를 친 후 python3를 입력합니다.

```
root@goorm:/workspace/PythonSeminar18(master)# python3
Python 3.6.5 (default, Sep  5 2018, 06:12:07)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

# 처음으로 시작하는 Python

---

대화형 인터프리터에서 사칙연산

+, -, /(나눗셈), \*(곱셈)

```
>>> 5+3
8
>>> 7-4
3
>>> 2/5
0.4
>>> 3*4
12
>>> █
```

# Hello, World!

---

보통 가장 처음 만들어보는 기본 예제이다.

- 컴파일러, 통합 개발 환경, 런타임 환경이 정상적으로 작동하는지 확인
- "The C Programming Language"에서 비롯한다.
  - C언어의 대표 교과서중 하나

# 처음으로 시작하는 Python

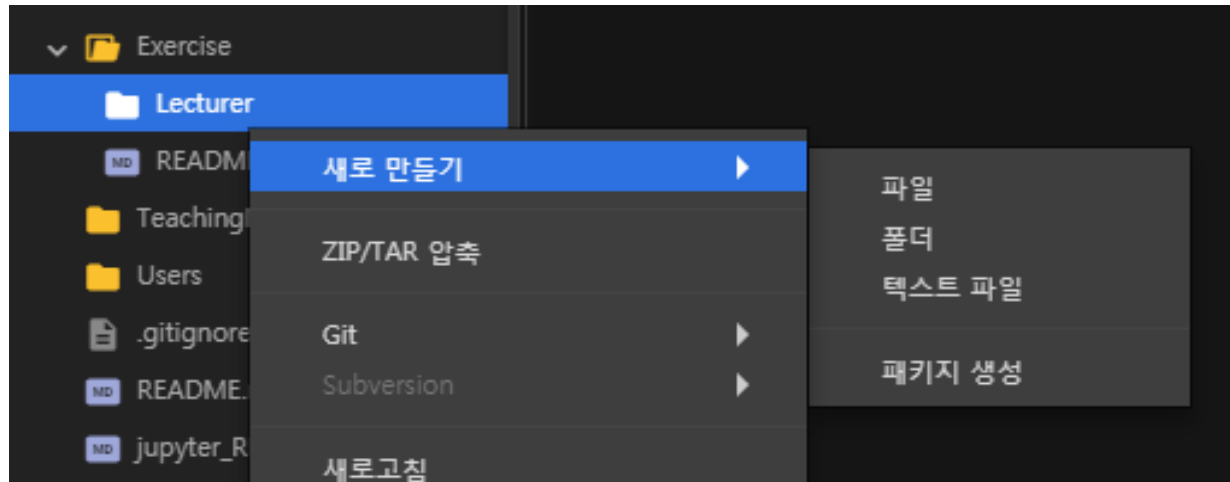
---

대화식 인터프리터 사용하기 - Hello, World!

```
>>> print("Hello, World!")  
Hello, World!  
>>> █
```

# 처음으로 시작하는 Python

hello\_world.py파일 만들기

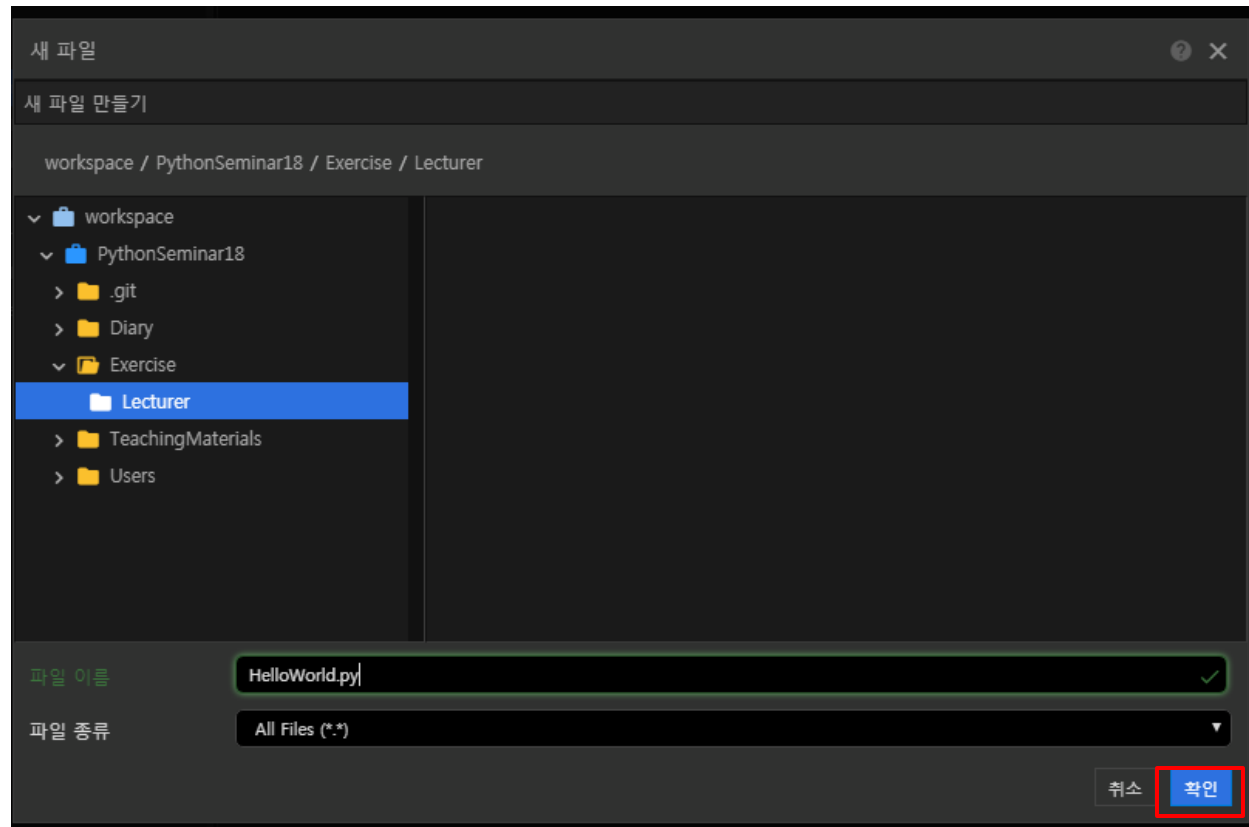


※ touch를 사용해도 된다.



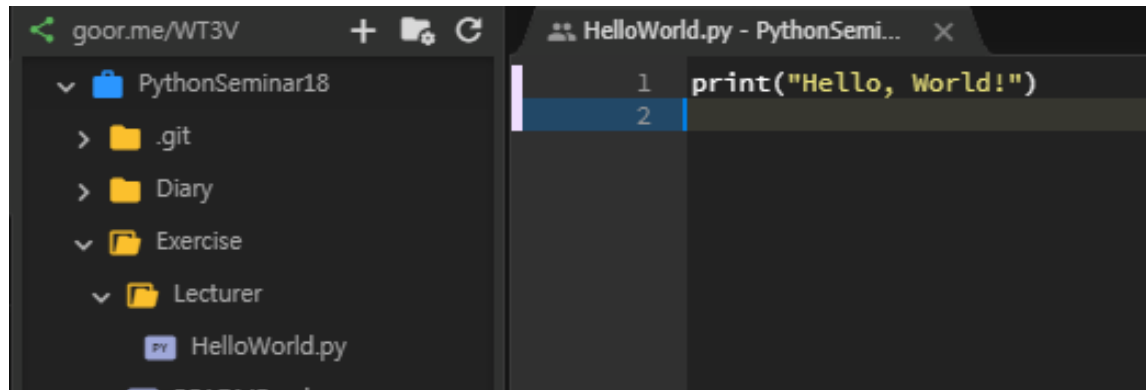
# 처음으로 시작하는 Python

hello\_world.py파일 만들기



# 처음으로 시작하는 Python

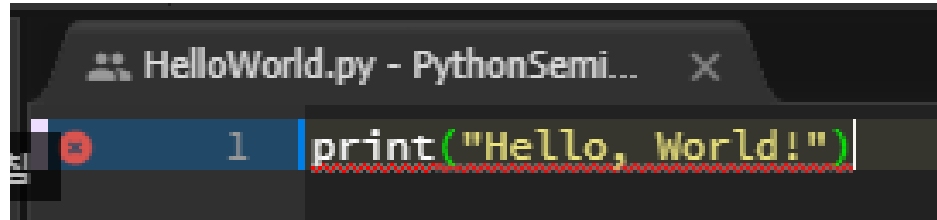
hello\_world.py파일 만들기



※ 코드에 색이 안 입혀지면 .py로 끝나는지 확인한다.

# 처음으로 시작하는 Python

---



```
HelloWorld.py - PythonSemi... X
1 print('Hello, World!')
```

※ 문법이 맞으나 빨간줄이 그어짐.

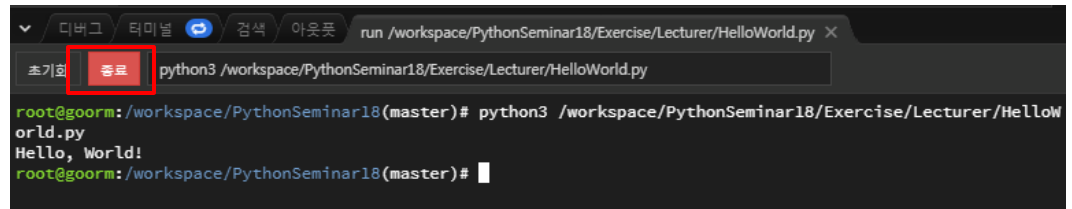
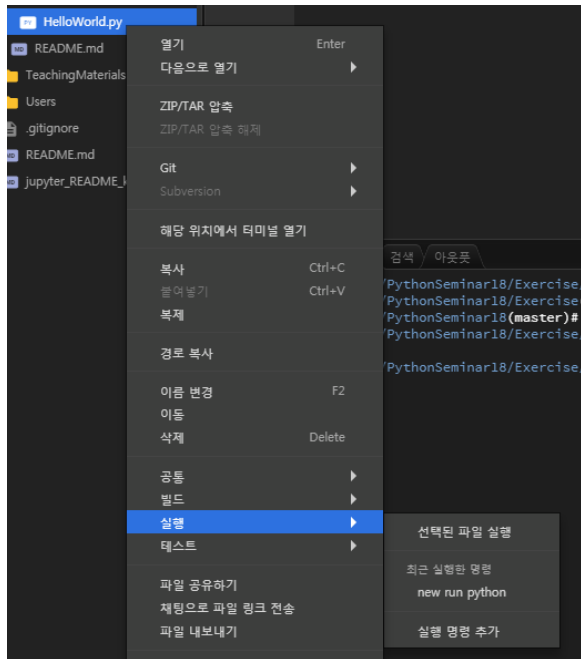
—파이썬의 코딩 컨벤션(pep8)

—무시하고 실행하거나 마지막 줄에 엔터를 하나 넣음

※ 빨간 x에 마우스를 올려두면 설명이 나온다.

# 처음으로 시작하는 Python

hello\_world.py파일 실행하기  
—구름 IDE 이용하기



# 처음으로 시작하는 Python

---

hello\_world.py파일 실행하기

- 터미널로 실행하기(추천)
- cd로 본인의 디렉터리로 이동한다.
- python3 실행파일명.py(tap키 가능)으로 실행한다.

```
root@goorm:/workspace/PythonSeminar18(master)# cd Exercise
root@goorm:/workspace/PythonSeminar18/Exercise(master)# ls -al
합계 28
drwxrwxr-x 5 root root 4096 12월 22 08:54 .
drwxrwxr-x 7 root root 4096 12월 22 08:52 ..
-rw-rw-r-- 1 root root 248 11월 27 05:29 README.md
drwxrwxr-- 2 root root 4096 12월 22 03:06 convert_fahrenheit_celsius
-rwxrwxr-x 1 root root 23 12월 8 00:52 hello_world.py
drwxrwxr-x 2 root root 4096 12월 15 08:23 lottery
drwxrwxr-- 2 root root 4096 12월 22 02:38 vending_machine
root@goorm:/workspace/PythonSeminar18/Exercise(master)# python3 hello_world.py
Hello, World!
```

# 기본과제

---

## 구름 IDE 가입하기

- 구름 IDE를 가입
- 공유된 컨테이너에 접속
- ./Users 아래에 자신의 별명으로 된 디렉토리를 생성
- 내부에 자신의 이름으로 파일을 생성

# 기본과제

---

## hello\_world.py파일생성

- hello\_world.py파일을 본인의 디렉터리에 생성
- 상위의 "처음으로 시작하는 Python"슬라이드를 참고하여 파일 내 내용을 입력한다.
- 파일을 실행하여 결과를 본다.