

프라이빗 클라우드 개발환경

나만의 서버 구축하기

리눅스 환경 개발환경 구축하기 (학생용)

• VPN서버 계정 생성

- VPN서버 담당자를 통해 계정 생성 (수업시간에 공지)

• VPN클라이언트 설치

- 윈도우OS: OpenVPN 클라이언트 다운로드 <https://openvpn.net/downloads/openvpn-connect-v3-windows.msi>
- 맥OS: TunnelBlick 클라이언트 다운로드 <https://tunnelblick.net/downloads.html>

• VPN클라이언트 세팅

- [XXX.ovpn 다운로드](#) (다운로드 암호는 mlclass20220302 / 기간 만료일 2022년 3월 20일)
- OpenVPN GUI 실행 >> 파일불러오기(xxx.ovpn) >> 설정
 - VPN계정 : ml_class, VPN 암호: mlclass20220302)

• VPN클라이언트 실행

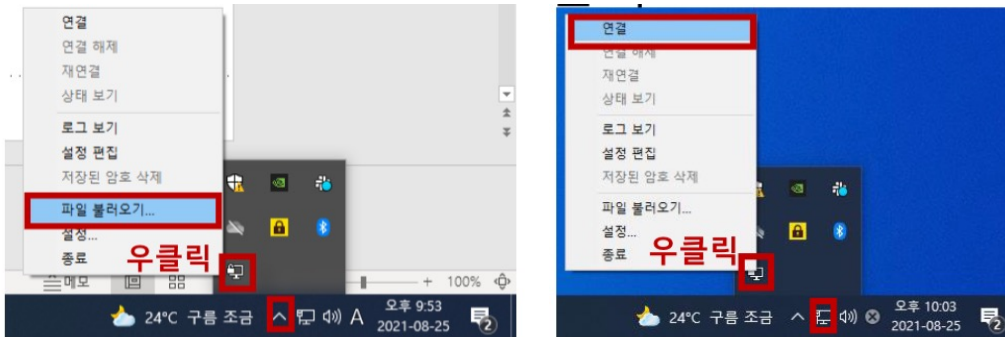
- OpenVPN GUI 실행 >> 연결



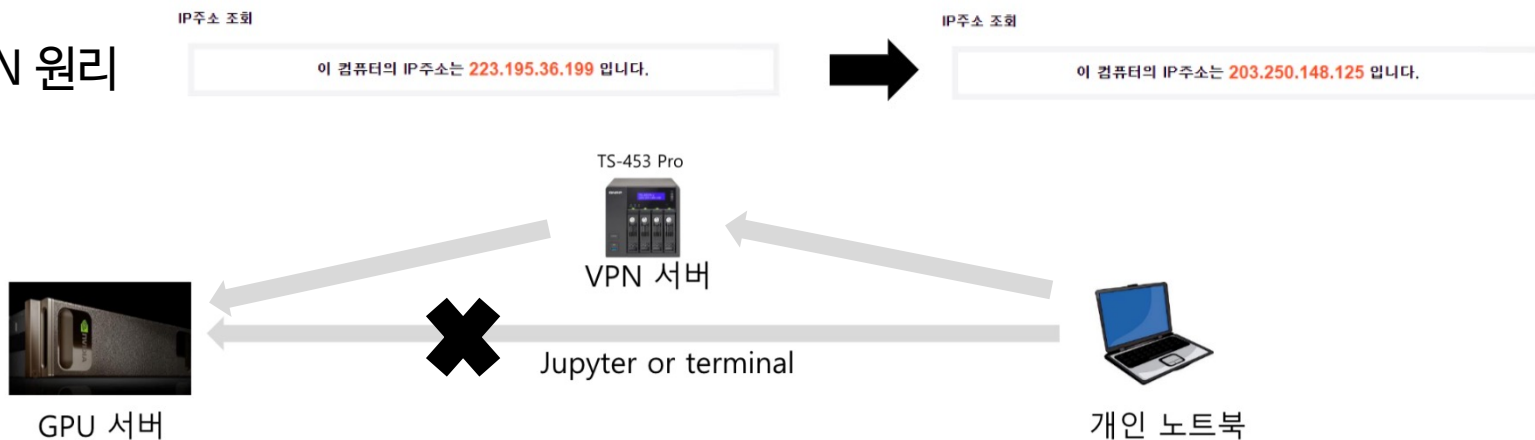
리눅스 환경 개발환경 구축하기 (학생용)

• VPN클라이언트 실행

- OpenVPN GUI 실행 >> 연결 >> VPN 계정 암호 입력



• VPN 원리



리눅스 환경 개발환경 구축하기 (학생용)

소문자 "엘"

- (자신의 터미널 환경) /# ssh -p 9001 -l root 203.250.148.128

```
ykchoi@YKChoi ~ % ssh -p 9001 -l root 203.250.148.128
root@203.250.148.128's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
```

```
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.
```

```
To restore this content, you can run the 'unminimize' command.
Last login: Sat Mar 12 03:26:57 2022 from 203.250.148.125
root@ec9b672c87da:~#
```

← 윈도우OS 사용자는 PowerShell 을 통해 명령어를 입력하시면 됩니다.

타이핑해야 하는 터미널 명령어 복사 붙여 넣거나, TAB 키를 활용하세요.

[[학생 개별 할당 ssh/web 포트 확인 링크](#)]

: PDF는 다운로드 받아야 링크를 클릭할 수 있음

* 9001 ← 9001이 아닌 (학생 개별적으로) 배정된 ssh 포트 입력하여 사용할 것

- root@(컨테이너 Hash ID):/# apt-get update
- root@(컨테이너 Hash ID):/# passwd
- [주의] root 암호를 변경 후 반드시 기억할 것
- root@(컨테이너 Hash ID):/# apt-get install net-tools
- root@(컨테이너 Hash ID):/# ifconfig -a

```
root@ec9b672c87da:~/workspace# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.4 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:ac:11:00:04 txqueuelen 0 (Ethernet)
    RX packets 207933 bytes 570325488 (570.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 95856 bytes 5403375 (5.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

[주의] 학생마다 할당 받은 컨테이너 IP 는 다르며, 주피터 노트북 설정을 위해 반드시 기억할 것

리눅스 환경 개발환경 구축하기 (학생용)

- root@(컨테이너 Hash ID):/# apt-get install wget
- root@(컨테이너 Hash ID):/# wget https://repo.anaconda.com/archive/Anaconda3-2020.11-Linux-x86_64.sh
- root@(컨테이너 Hash ID):/# bash ./Anaconda3-2020.11-Linux-x86_64.sh
 - ➔ 스페이스 바를 통해 제시된 라이선스 관련 글을 읽고, 의견을 묻는 곳에서는 “YES” 를 타이핑

- root@(컨테이너 Hash ID):/# source ~/.bashrc

```
root@ec9b672c87da:~/workspace# source ~/.bashrc
(base) root@ec9b672c87da:~/workspace#
```

- (base) root@(컨테이너 Hash ID):/# conda update conda
- (base) root@(컨테이너 Hash ID):/# conda update anaconda
- (base) root@(컨테이너 Hash ID):/# conda update -n base conda
- (base) root@(컨테이너 Hash ID):/# apt install python3-pip
- (base) root@(컨테이너 Hash ID):/# conda create -n ml_class python=3.7 numpy scipy matplotlib spyder pandas seaborn scikit-learn h5py statsmodels
- (base) root@(컨테이너 Hash ID):/# conda activate ml_class

```
(base) root@ec9b672c87da:~# conda activate ml_class
(ml_class) root@ec9b672c87da:~#
```

- (ml_class) root@(컨테이너 Hash ID):/# conda deactivate

```
(ml_class) root@ec9b672c87da:~# conda deactivate
root@ec9b672c87da:~#
```

리눅스 환경 개발환경 구축하기 (학생용)

- root@(컨테이너 Hash ID):/# conda activate ml_class
- (ml_class) root@(컨테이너 Hash ID):/# conda install jupyter notebook
- (ml_class) root@(컨테이너 Hash ID):/# jupyter-notebook --generate-config
- (ml_class) root@(컨테이너 Hash ID):/# ipython

```
(ml_class) root@ec9b672c87da:~# ipython
Python 3.7.11 (default, Jul 27 2021, 14:32:16)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.29.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from notebook.auth import security

In [2]: security.passwd()
Enter password:
Verify password:
Out[2]: 'argon2:$argon2id$v=19$m=10240,t=10,p=8$MEX33T90vDc/mJs1EqhwKg$zMbNjolQ0eqI3KBkpzoAkB0+KueViIITPYoUqzyJ+Cc'
```

* 주피터 노트북 로그인을 위한 암호 설정

* 주피터 노트북 설정을 위한 암호화 키 메모장에 복사하여 보관

- (ml_class) root@(컨테이너 Hash ID):/# vi ~/.jupyter/jupyter_notebook_config.py

Vi 에서 키보드 입력을 원할 때: i

```
# Configuration file for jupyter-notebook.

c = get_config()
c.JupyterApp.config_file_name = 'jupyter_notebook_config.py'
c.NotebookApp.allow_origin = '*'
c.NotebookApp.ip = '172.17.0.4'
c.NotebookApp.open_browser = False
c.NotebookApp.password = u'argon2:$argon2id$v=19$m=10240,t=10,p=8$MEX33T90vDc/mJs1EqhwKg$zMbNjolQ0eqI3KBkpzoAkB0+KueViIITPYoUqzyJ+Cc'
c.NotebookApp.allow_root = True
c.NotebookApp.port = 8001
```

Vi 에서 저장하고 나갈 때 :wq

* c.NotebookApp.ip = 'xxx.xxx.xxx.xxx' ◀ 할당 받은 컨테이너 IP 로 입력

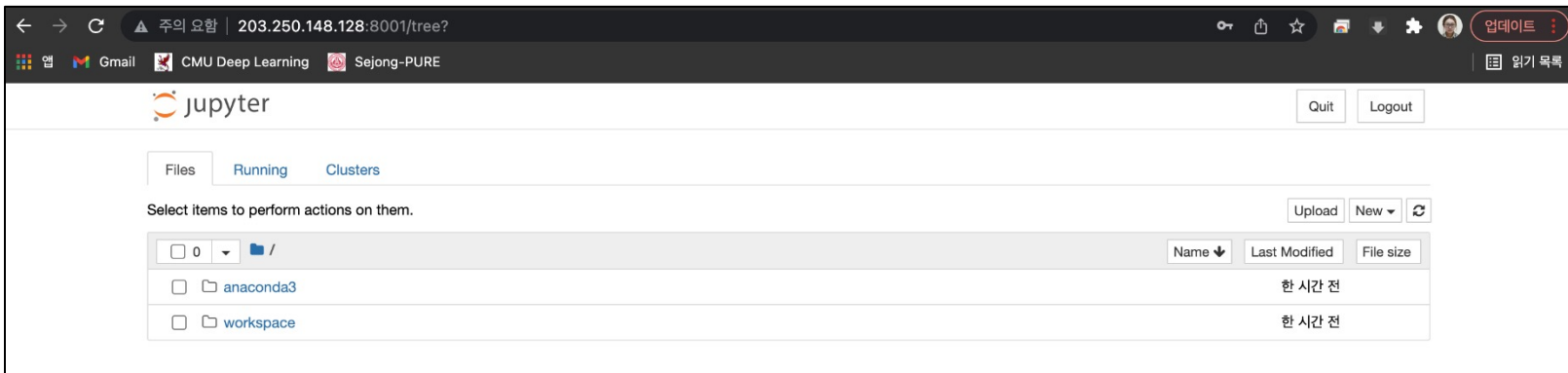
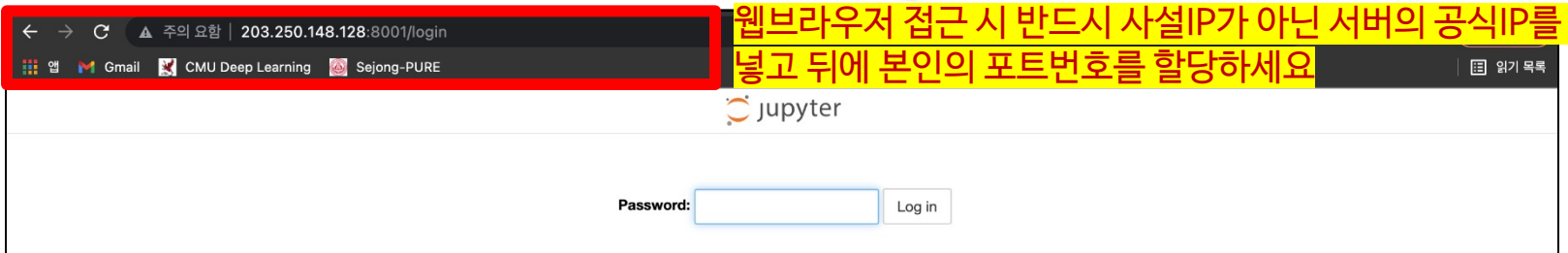
* c.NotebookApp.password = u'xxxxxxxxxxxxxxxx' ◀ 주피터 노트북 암호화 키 붙여넣기

* c.NotebookApp.port = xxxx ◀ (학생 개별적으로) 배정된 web 포트 입력

리눅스 환경 개발환경 구축하기 (학생용)

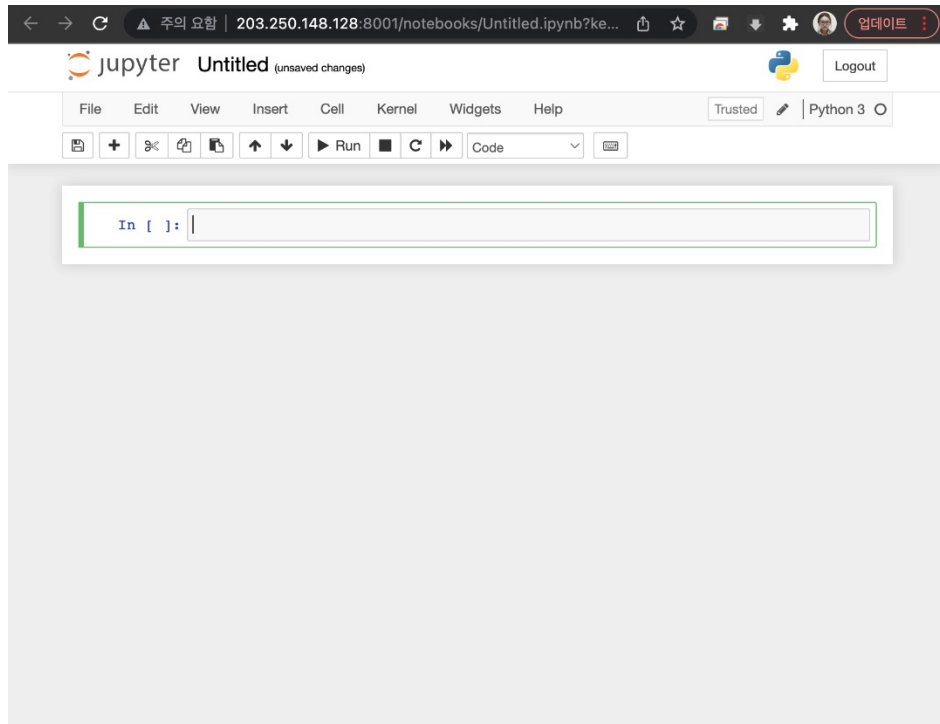
➤ (ml_class) root@(컨테이너 Hash ID):/# jupyter notebook

```
(ml_class) root@ec9b672c87da:~# jupyter notebook
[I 01:54:50.899 NotebookApp] Serving notebooks from local directory: /root
[I 01:54:50.899 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 01:54:50.899 NotebookApp] http://172.17.0.4:8001/
[I 01:54:50.899 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[I 01:55:42.742 NotebookApp] 302 GET / (203.250.148.125) 1.440000ms
[W 01:55:42.796 NotebookApp] Clearing invalid/expired login cookie username-203-250-148-128-8001
[I 01:55:42.798 NotebookApp] 302 GET /tree? (203.250.148.125) 3.150000ms
[I 01:55:45.581 NotebookApp] 302 POST /login?next=%2Ftree%3F (203.250.148.125) 70.190000ms
```

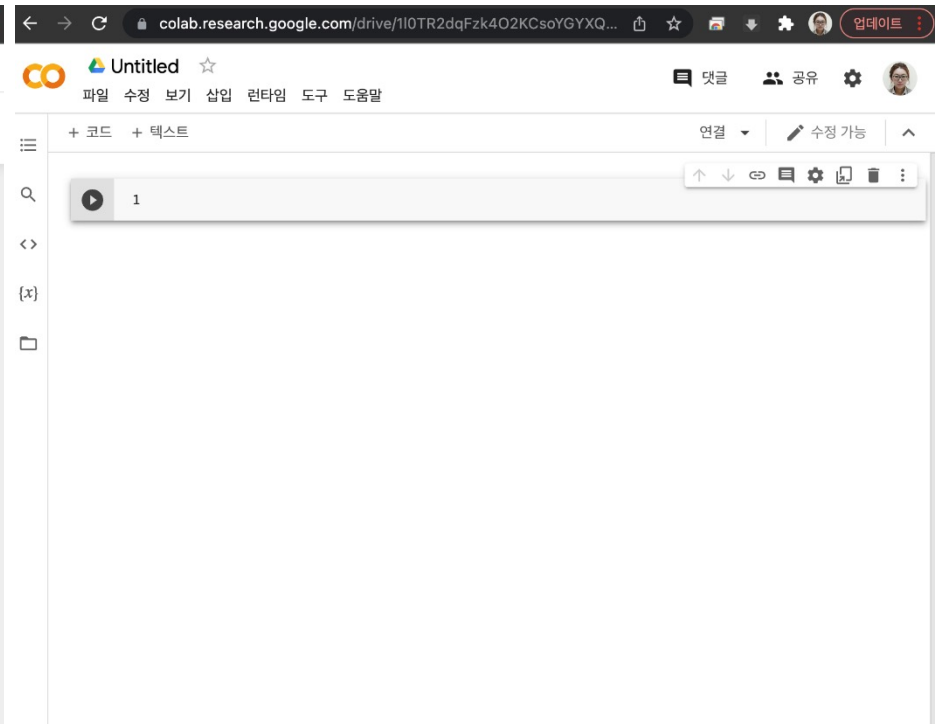


리눅스 환경 개발환경 구축하기 (학생용)

➤ (ml_class) root@(컨테이너 Hash ID):/# jupyter notebook



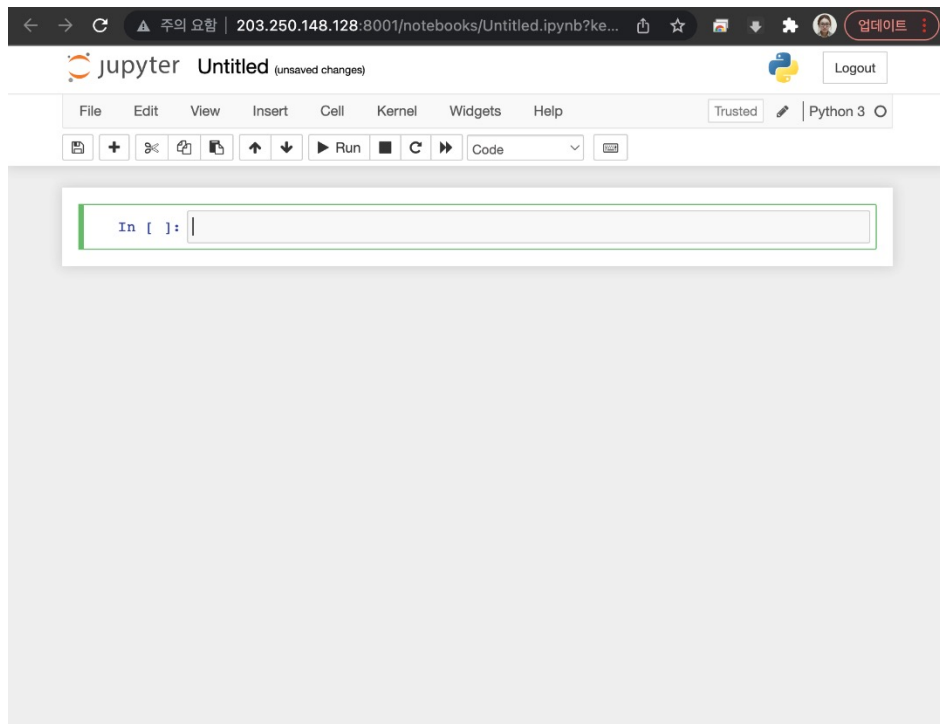
직접 구축한 클라우드 기반의 주피터 노트북 개발환경



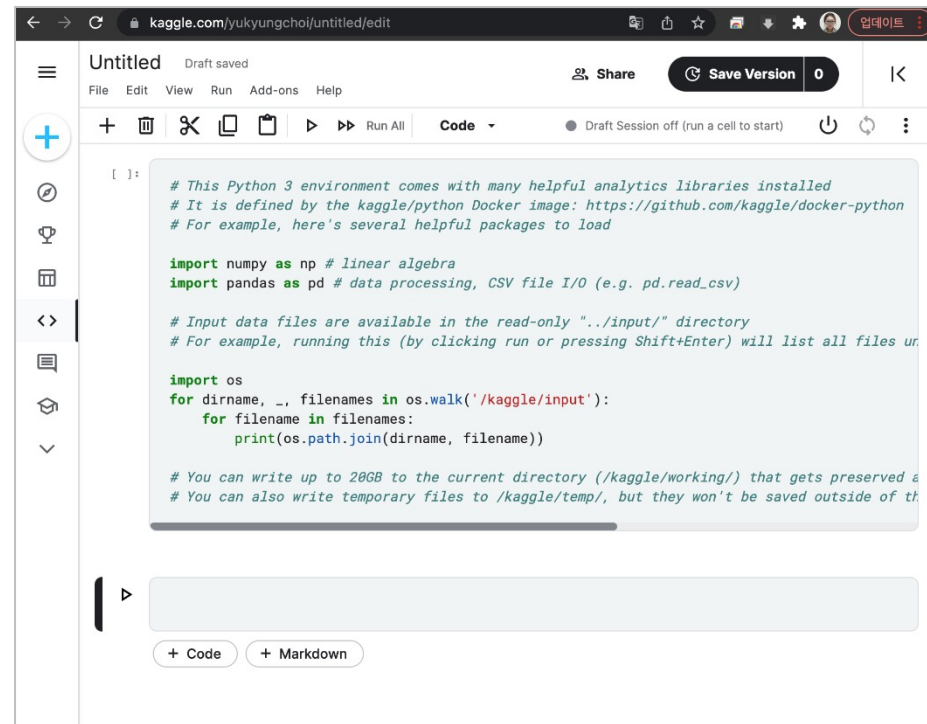
코랩에서 제공받은 클라우드 기반의 주피터 노트북 개발환경

리눅스 환경 개발환경 구축하기 (학생용)

➤ (ml_class) root@(컨테이너 Hash ID):/# jupyter notebook



직접 구축한 클라우드 기반의 주피터 노트북 개발환경



캐글에서 제공받은 클라우드 기반의 주피터 노트북 개발환경

리눅스 환경 개발환경 구축하기 (학생용)

- (ml_class) root@(컨테이너 Hash ID):/# nohub jupyter notebook &

```
(base) root@ec9b672c87da:~# nohub jupyter notebook &
[1] 10218
(base) root@ec9b672c87da:~# nohub: ignoring input and appending output to 'nohub.out'
█
```

- (ml_class) root@(컨테이너 Hash ID):/# fg %1

```
(base) root@ec9b672c87da:~# ps -al
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   0 10218 10181  1  80   0 - 15187 ep_pol pts/3    00:00:00 jupyter-noteboo
0 R   0 10222 10181  0  80   0 - 1891 -      pts/3    00:00:00 ps
(base) root@ec9b672c87da:~# fg %1
nohub jupyter notebook
█
```

- (ml_class) root@(컨테이너 Hash ID):/# ctrl+c

```
nohub jupyter notebook
^C(base) root@ec9b672c87da:~# ps -al
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 R   0 10223 10181  0  80   0 - 1891 -      pts/3    00:00:00 ps
(base) root@ec9b672c87da:~# █
```

pdb를 이용한 디버깅

- pdb 모듈은 파이썬 코드의 디버깅(Debugging)을 위해 사용되는 모듈
 - 디버깅이란? 코드에서 버그를 제거하는 것을 의미함
 - 중단점(breakpoint) 설정, 소스 리스팅, 변수 치환, 콜스택 검사 지원

```
1 import pdb
2
3 numbers = [1, 2, 3]
4 for i in range(len(numbers)):
5     pdb.set_trace() # 중단점 설정
6     if numbers[i] % 2 == 0:
7         del numbers[i]
8
9 print(numbers)
```

- pdb 모듈 기본 사용법
 - 모듈 사용을 위한 import pdb
 - 중단점 설정을 위한 pdb.set_trace()
 - (pdb) 프롬프트에서 아래의 명령 수행 가능
 - Continue의 **c**를 입력: 다음 중단점으로 이동, 중단점 없으면 끝까지 실행
 - Next의 **n**을 입력: 다음 라인으로 이동
 - List의 **l**을 입력: 소스코드에서 현재의 진행 위치를 출력

pdb를 이용한 디버깅

- [과제] 파이썬 코드 디버깅
 - (원 코드) → (수정된 코드: 제출)

```
1 import pdb
2
3 numbers = [1, 2, 3]
4 for i in range(len(numbers)):
5     pdb.set_trace() # 중단점 설정
6     if numbers[i] % 2 == 0:
7         del numbers[i]
8
9 print(numbers)
```