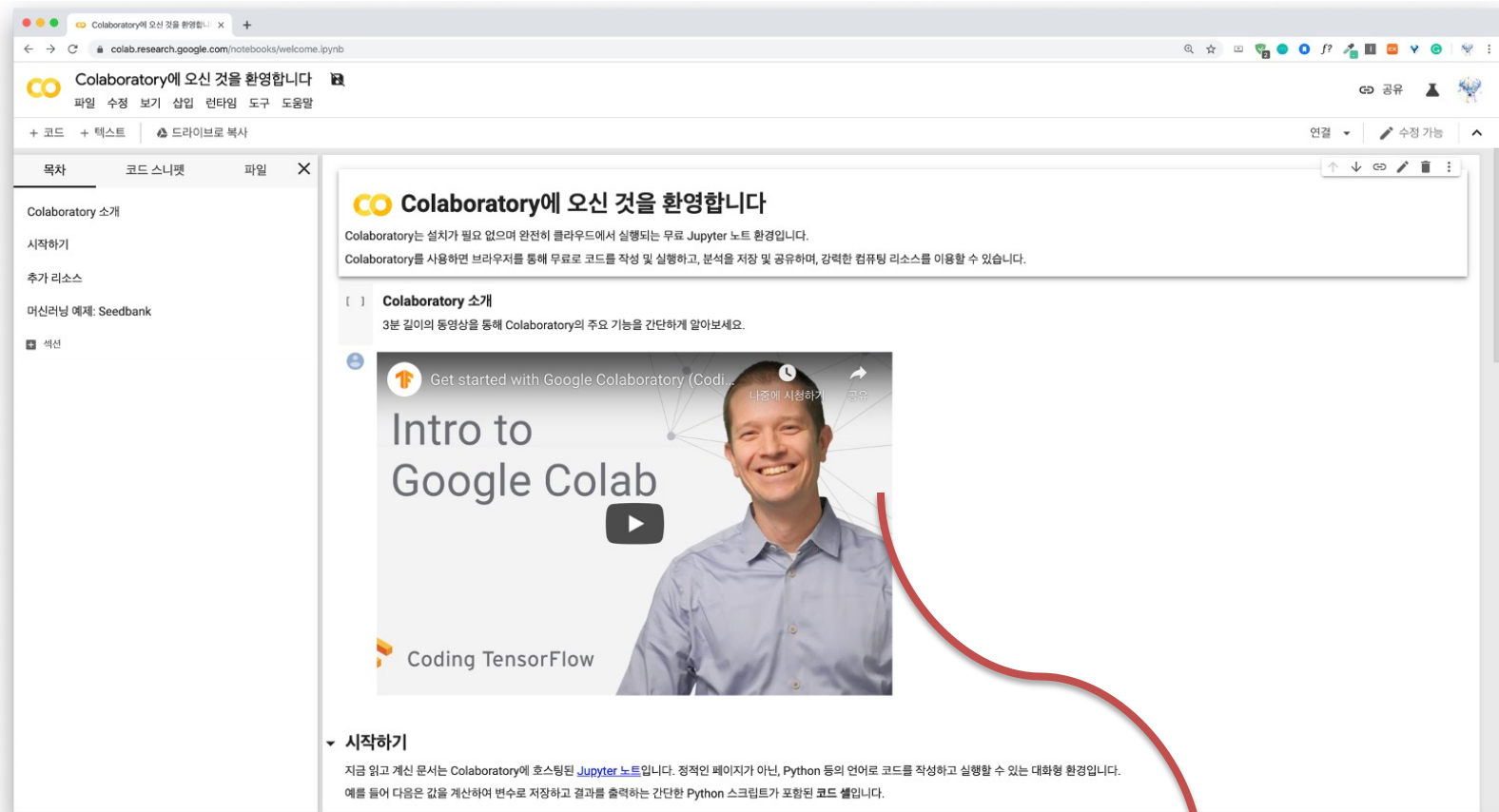


퍼블릭 클라우드 개발환경

구글 코랩 (Colab)

코랩 실습 환경 세팅

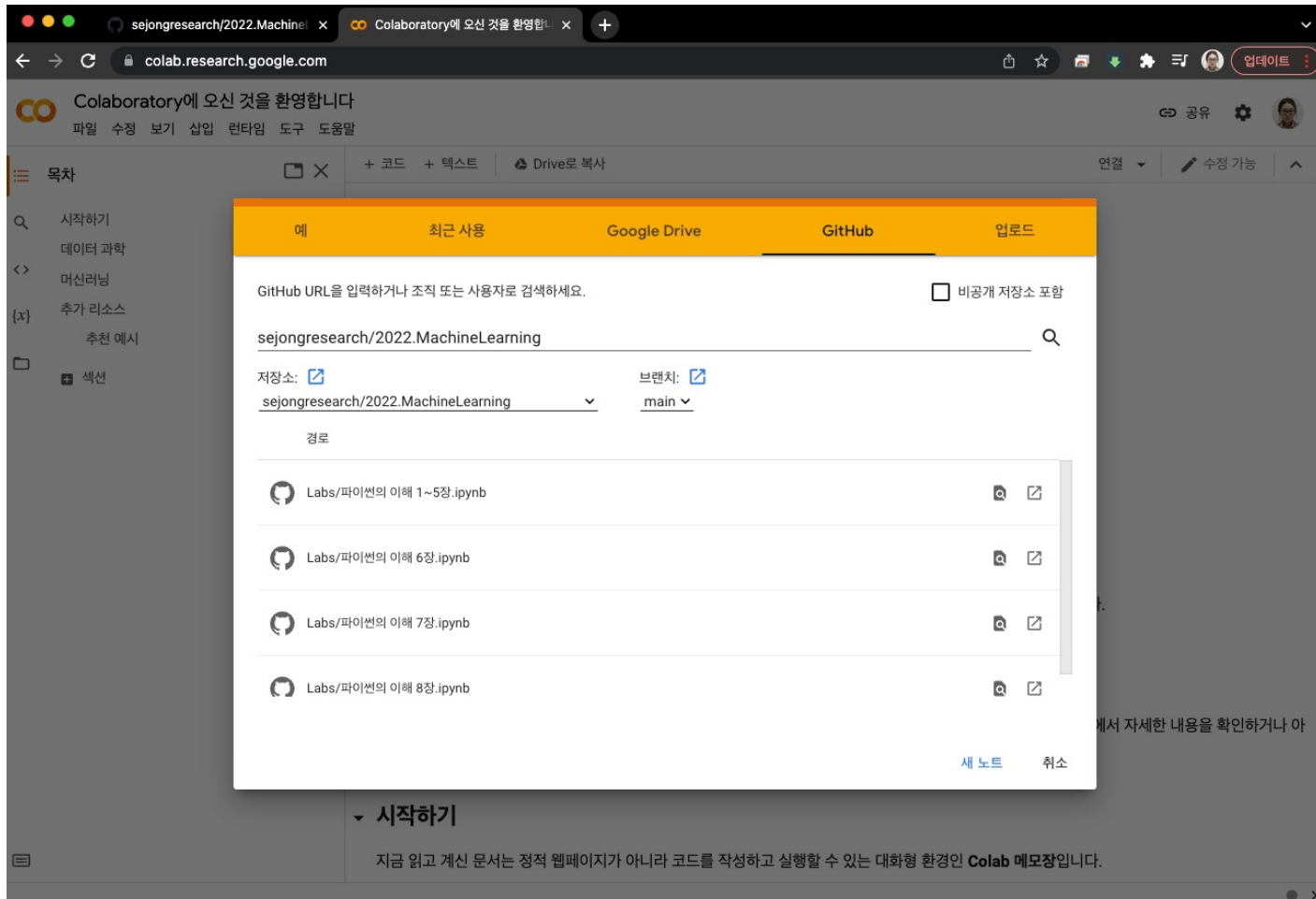
- 코랩 공식 링크: <https://colab.research.google.com>



<https://www.youtube.com/watch?v=inN8seMm7UI&feature=youtu.be>

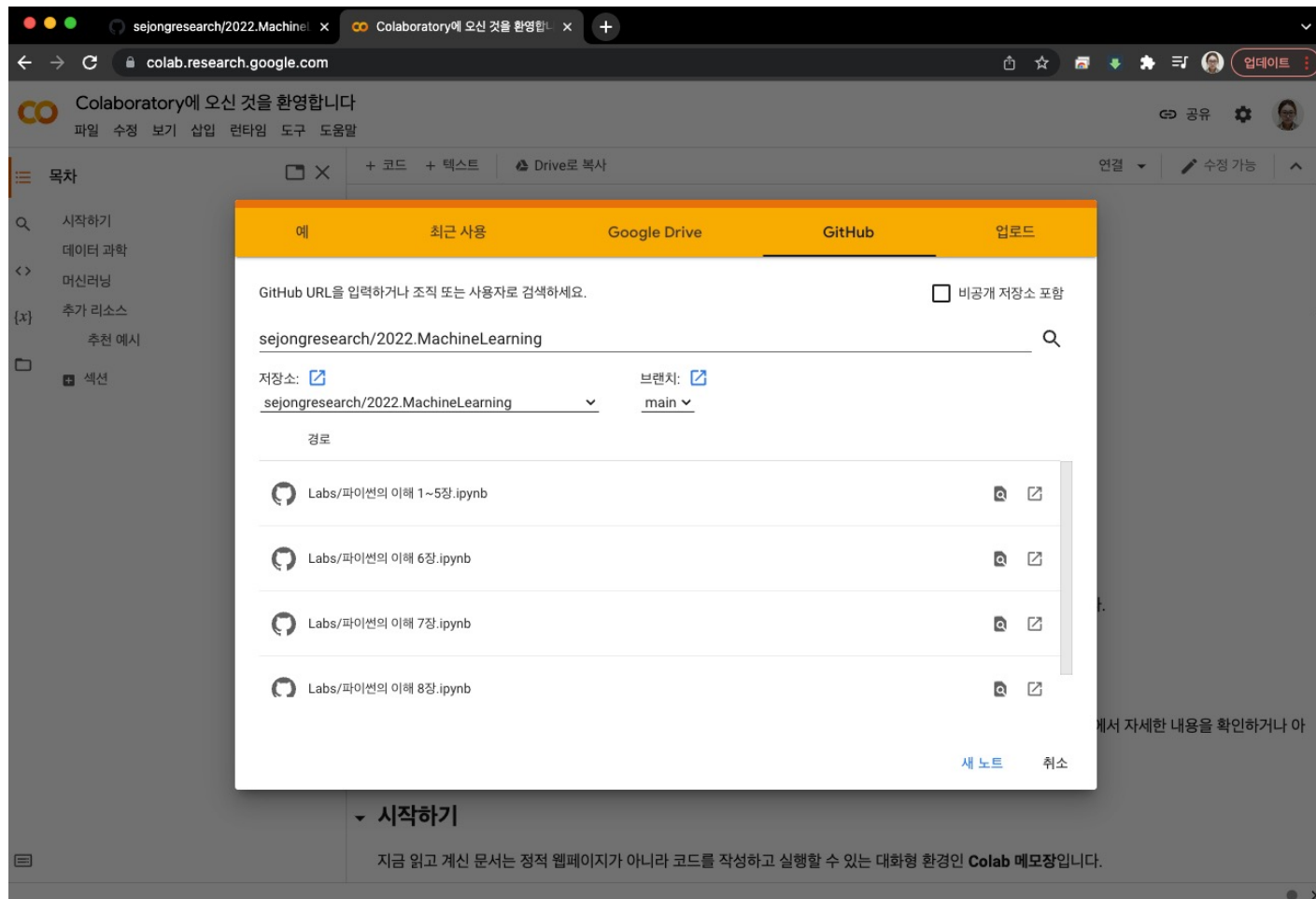
코랩 실습 환경 세팅

- 파일 → 노트 열기 → GitHub탭 → sejongresearch/2022.MachineLearning



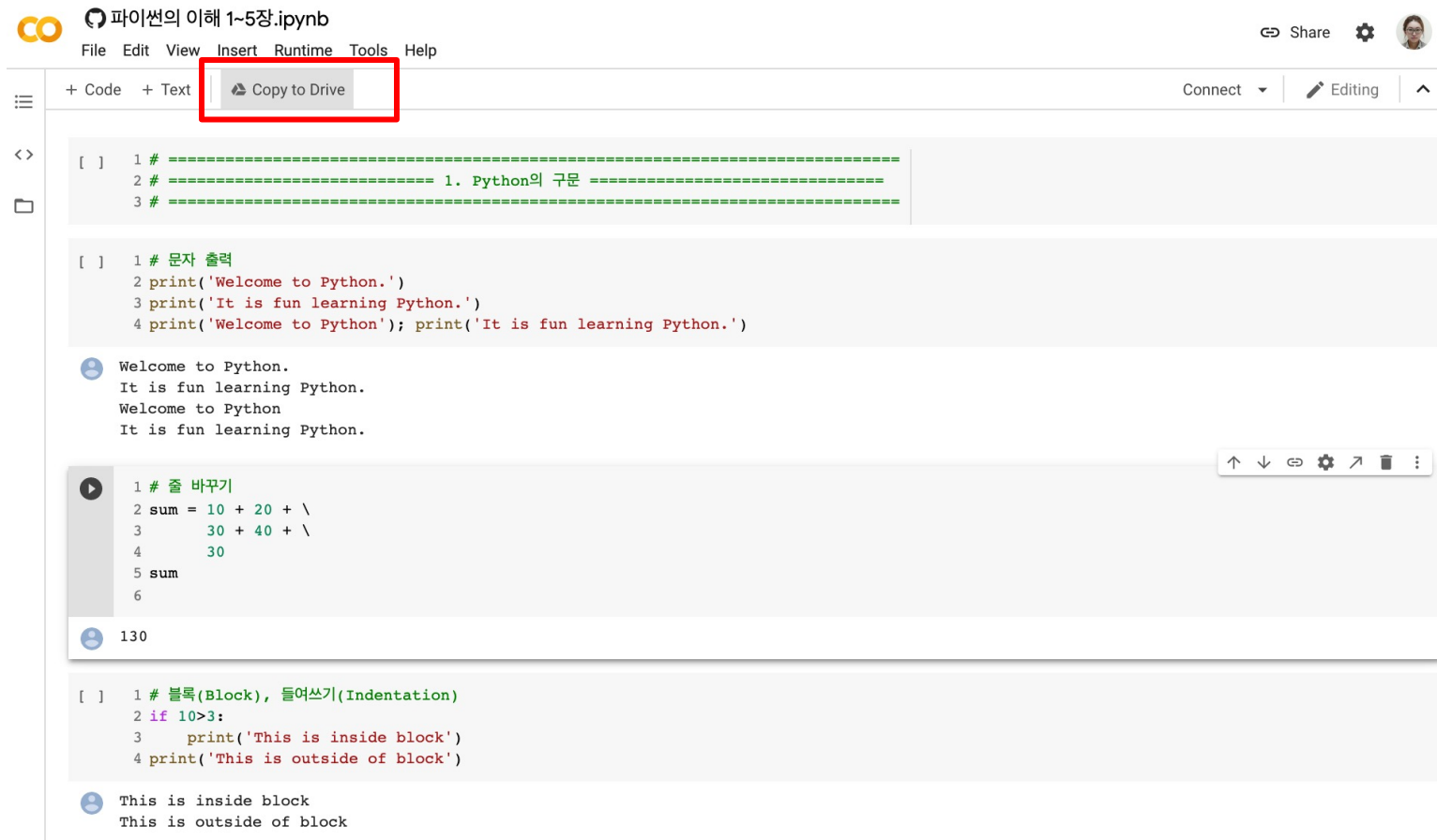
코랩 실습 환경 세팅

▪ Labs/파이썬의 이해 1~5장.ipynb 열기



코랩 실습 환경 세팅

- [Copy to Drive] 클릭
 - 반드시 사용자의 개인 계정 구글 드라이브로 복사 후 실행해야 함



The screenshot shows the Google Colab interface for a file named '파이썬의 이해 1~5장.ipynb'. The top menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu, there are buttons for '+ Code', '+ Text', and 'Copy to Drive', with the 'Copy to Drive' button highlighted by a red rectangle. To the right of these buttons are 'Connect', 'Editing', and a user profile icon. The main area displays three code blocks with their corresponding outputs. The first block contains a comment about Python syntax. The second block contains a multi-line print statement, and its output shows the text 'Welcome to Python.' and 'It is fun learning Python.' printed twice. The third block contains a code snippet for calculating a sum, and its output shows the value '130'. The bottom block contains a code snippet for testing block indentation, and its output shows 'This is inside block' and 'This is outside of block'.

파이썬의 이해 1~5장.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text **Copy to Drive** Connect Editing

```
[ ] 1 # =====
2 # ===== 1. Python의 구문 =====
3 # =====
```

```
[ ] 1 # 문자 출력
2 print('Welcome to Python.')
3 print('It is fun learning Python.')
4 print('Welcome to Python'); print('It is fun learning Python.')
```

Welcome to Python.
It is fun learning Python.
Welcome to Python
It is fun learning Python.

```
▶ 1 # 줄 바꾸기
2 sum = 10 + 20 + \
3     30 + 40 + \
4     30
5 sum
6
```

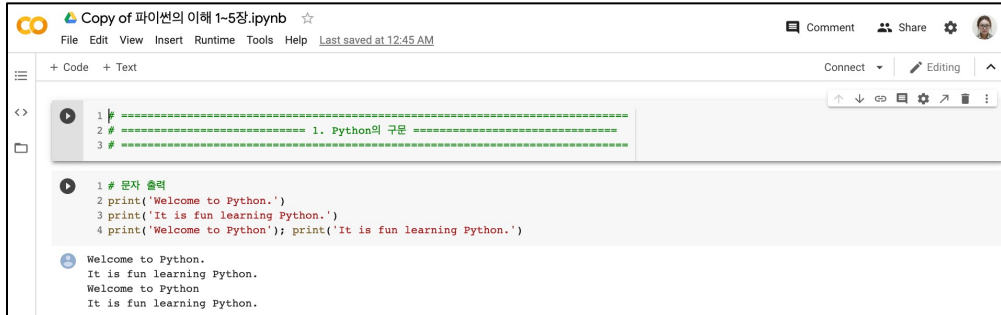
130

```
[ ] 1 # 블록(Block), 들여쓰기(Indentation)
2 if 10>3:
3     print('This is inside block')
4 print('This is outside of block')
```

This is inside block
This is outside of block

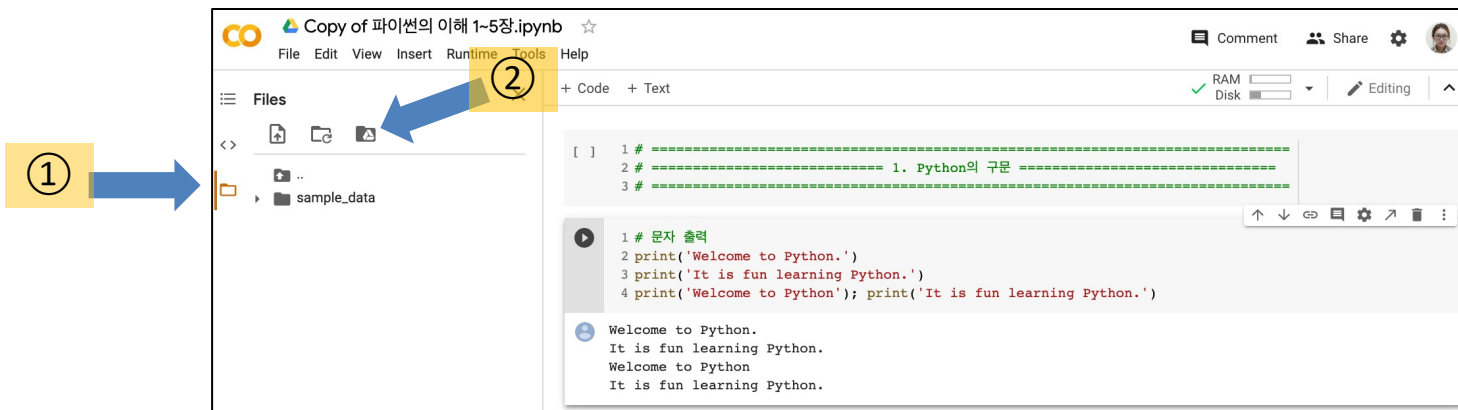
코랩 사용법

- Cell 실행하는 법
 - [Play 버튼 클릭] 혹은 [Shift + Enter]



- Google Drive 연동하는 법 (*개인 클라우드 계정을 사용해야 파일 영구 저장됨)

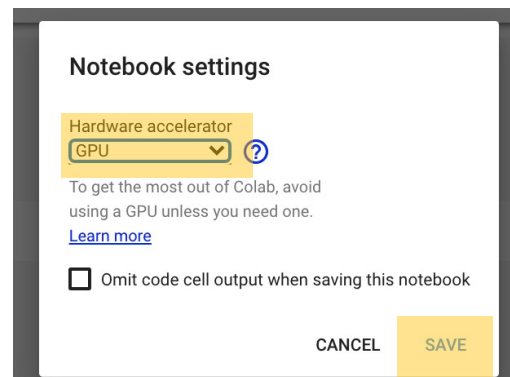
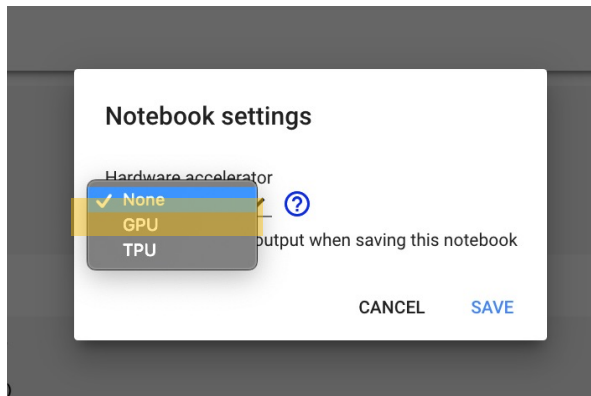
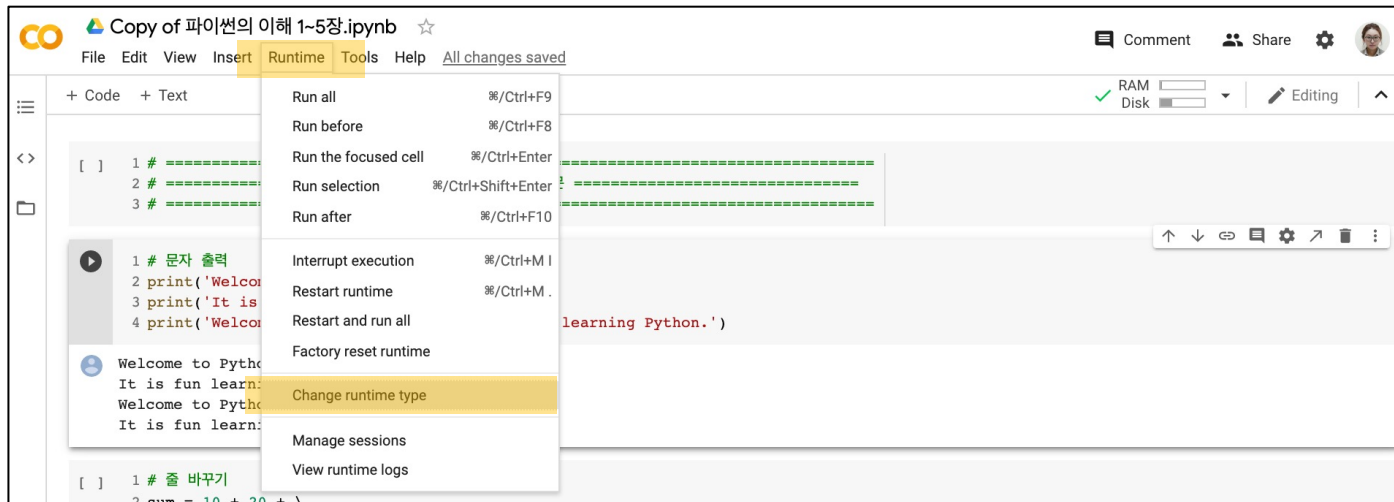
- ① → ② 클릭



코랩 사용법

- GPU 사용을 위한 런타임 설정

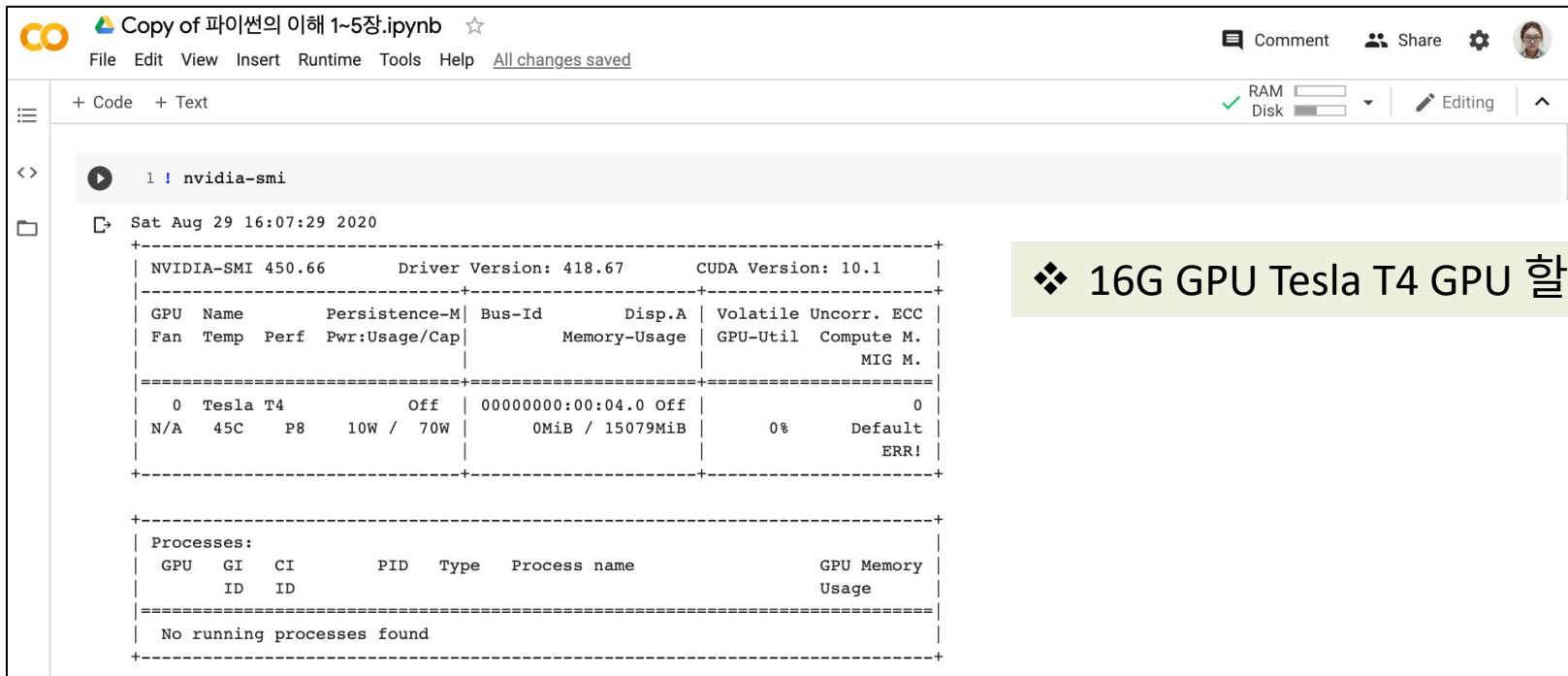
- Runtime → Change runtime type → Hardware accelerator → GPU → Save



코랩 사용법

■ 코랩 개발 환경 확인 방법

- 코랩 은 Cell 안에서 ! [리눅스 명령어] 를 통한 셀 실행을 지원함
- GPU 개발 환경 확인을 위한 nvidia-smi 명령어 활용



```
Copy of 파이썬의 이해 1~5장.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
1 ! nvidia-smi

Sat Aug 29 16:07:29 2020

+-----+
| NVIDIA-SMI 450.66                Driver Version: 418.67          CUDA Version: 10.1         |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A    Volatile Uncorr. ECC  |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage  GPU-Util  Compute M.  |
|                                           MIG M.         |
+-----+-----+
|    0   Tesla T4              Off   | 00000000:00:04:0 Off    |          0          |
| N/A   45C    P8      10W /  70W   |    0MiB / 15079MiB      0%      Default  |
|                                           ERR!              |
+-----+-----+

Processes:
+-----+
| GPU   GI    CI          PID    Type    Process name                      GPU Memory |
| ID    ID                                     Name                                Usage      |
+-----+
| No running processes found          |
+-----+
```

❖ 16G GPU Tesla T4 GPU 할당됨

코랩 사용법

■ 코랩 개발 환경 확인 방법

- CPU 스펙 확인을 위한 `cpuinfo` 명령어 실행
- 메모리 사용/미사용 정보 확인을 위한 `free` 명령어 실행
- 서버 운영체제 확인법 실행

```
+ Code + Text
1 !cat /proc/cpuinfo

processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 79
model name    : Intel(R) Xeon(R) CPU @ 2.20GHz
```

❖ Xeon CPU 2개 할당됨

```
+ Code + Text
[3] 1 !free -h

             total        used        free      shared  buff/cache   available
Mem:          12G          535M          10G          972K          2.0G          11G
Swap:           0B           0B           0B
```

❖ 메모리 12G 할당됨

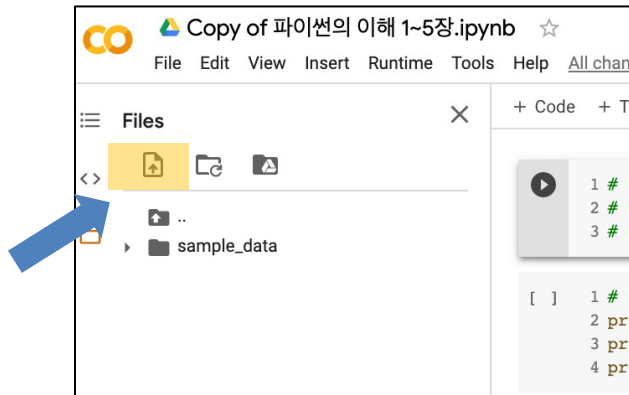
```
+ Code + Text
[1] 1 import platform
    2 platform.platform()

'Linux-4.19.112+-x86_64-with-Ubuntu-18.04-bionic'
```

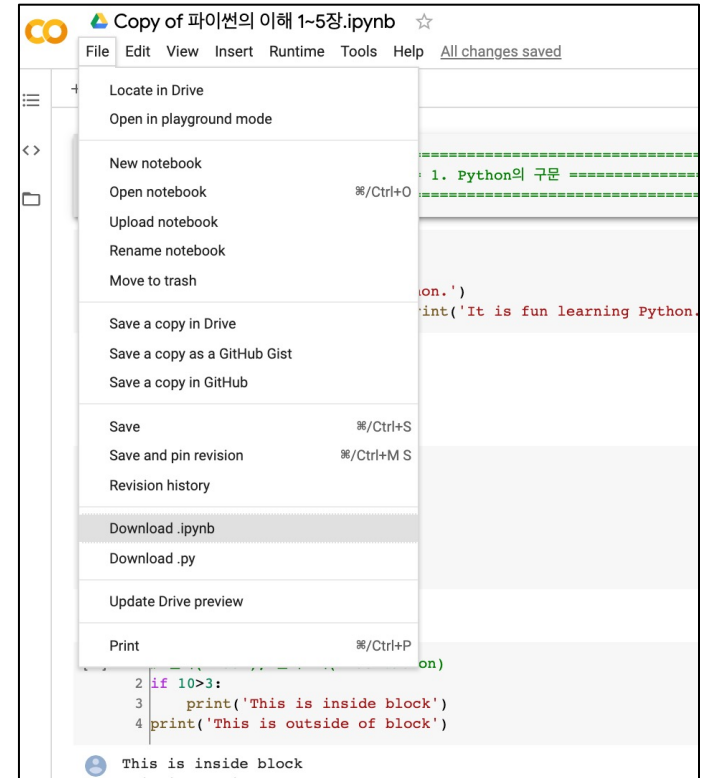
❖ 리눅스 18.04 운영체제

코랩 사용법

- 파일 업로드 하기
 - 업로드 버튼 클릭 ➔ 파일 선택



- 작성한 소스 파일 다운로드 하기
 - File ➔ Download .ipynb or .py



코랩 사용법: 마크다운 주석

■ [1단계] 헤더 (Header)

- 제목, 문단별 제목을 쓰고 싶을 때, 글의 구조(개요) 및 큰 틀을 잡을 때 사용한다.

```
# 제목 1단계
## 제목 2단계
### 제목 3단계
#### 제목 4단계
##### 제목 5단계
##### 제목 6단계
```

제목 1단계

제목 2단계

제목 3단계

제목 4단계

제목 5단계

제목 6단계

■ [2단계] 목록 (List)

- 요소를 나열 할 때

```
1. 첫번째
1. 두번째
1. 세번째

+ 순서없음
- 홍길동
  * 중대장
  + 프로실망러
```

1. 첫번째

2. 두번째

3. 세번째

- 순서없음
 - 홍길동
 - 중대장
 - 프로실망러

코랩 사용법: 마크다운 주석

■ [3단계] 수평선

- 내용을 명시적으로 구분하고 싶을 때

■ [4단계] 강조

- 문장 내 강조하고 싶은 단어를 눈에 띄게

__볼드(진하게)__
이탤릭체(기울여서)
~~취소선~~

<u>밑줄</u>

__볼드로 진하게 만들다가*이탤릭으로 기울이고*다시 볼드로__(중복 활용도 가능하다.)

볼드(진하게)

이탤릭체(기울여서)

취소선

밑줄

볼드로 진하게 만들다가이탤릭으로 기울이고다시 볼드로(중복 활용도 가능하다.)

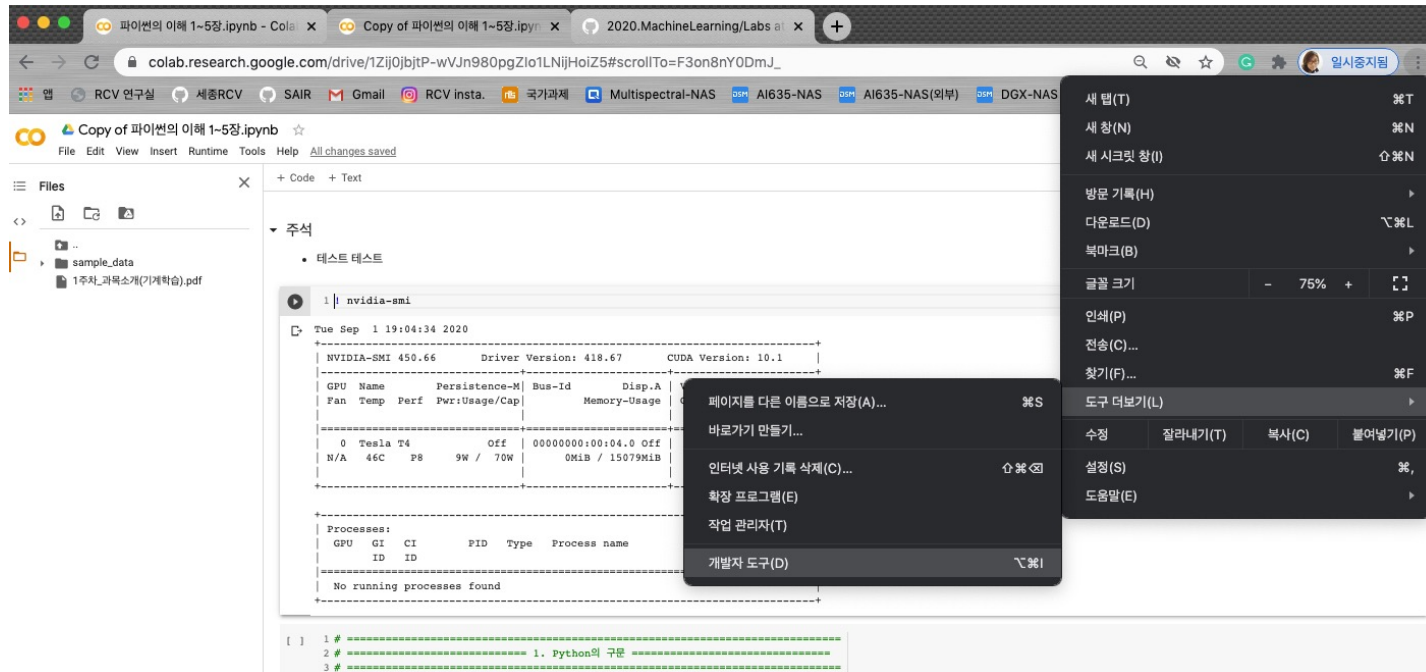
코랩 사용법

■ 코랩 세션의 단점

- 한 세션의 최대 유지 가능 시간은 **12시간**이지만 **90분** 이상 작업이 없는 경우 강제 세션 종료됨

■ 세션 유지 방법

- 브라우저의 개발자 모드(F12)의 console 탭을 통해서 실행시키면 된다.



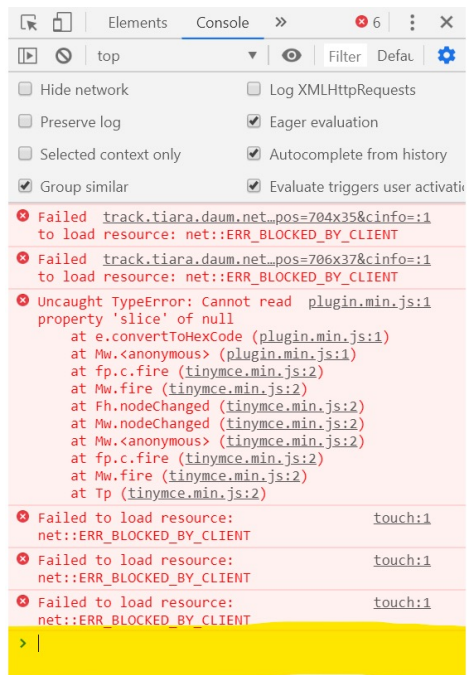
코랩 사용법

■ 코랩 세션의 단점

- 한 세션의 최대 유지 가능 시간은 12시간이지만 90분 이상 작업이 없는 경우 강제 세션 종료됨

■ 세션 유지 방법

- 브라우저의 개발자 모드(F12)의 console 탭을 통해서 실행시키면 된다.



코랩 사용법

- 구글 colab에서 90 time out 세션 유지 javascript 코드

```
function ClickConnect() {  
    var buttons = document.querySelectorAll("colab-dialog.yes-no-dialog paper-  
button#cancel");  
    buttons.forEach(function(btn) {  
        btn.click();  
    });  
    console.log("1분마다 자동 재연결");  
    document.querySelector("colab-toolbar-button#connect").click();  
}  
setInterval(ClickConnect,1000*60);
```

코랩 사용법

- 구글 colab에서 buffered data was truncated after reaching the output size limit 에러 방지를 위한 현재 출력 창 자동 지우기

```
function CleanCurrentOutput(){
    var btn = document.querySelector(".output-icon.clear_outputs_enabled.output-icon-selected[title$='현재 실행 중...'] iron-icon[command=clear-focused-or-selected-outputs]");
    if(btn) { console.log("30분마다 출력 지우기");
        btn.click();
    }
}
setInterval(CleanCurrentOutput,1000*60*30);
```


코랩 사용법

- 캐글 API를 이용해 코랩 사용하기

```
function CleanCurrentOutput(){
    var btn = document.querySelector(".output-icon.clear_outputs_enabled.output-icon-selected[title$='현재 실행 중...'] iron-icon[command=clear-focused-or-selected-outputs]");
    if(btn) { console.log("30분마다 출력 지우기");
        btn.click();
    }
}
setInterval(CleanCurrentOutput,1000*60*30);
```