

# 개요

파이썬을 활용한 머신러닝의 첫 관문! 구글 Colab(코랩)의 환경구성 및 기본 사용법을 다룬 포스트입니다.

- 목차
  - Colab이란 무엇인가?
  - Colab 환경설정
  - Colab을 활용한 간단한 예제 작성
  - Colab & Markdown

## Colab이란 무엇인가?

구글 코랩(Colab)은 클라우드 기반의 무료 Jupyter 노트북 개발 환경이다. 내부적으로는 코랩 + 구글드라이브 + 도커 + 리눅스 + 구글클라우드의 기술스택으로 이루어진 것으로 알려져있다.

흔히들 딥러닝은 귀족학문이라고 말한다. 대학원에서 등록금을 지불해야 함은 물론, 꽤 긴 시간의 고찰과 연구를 필요로 하며(그 시간동안 돈을 못벌다. 오히려 더 써야 할지도...), 비싼 Machine을 가진 자가 더 빠른 결과를 얻을 수 있다. 금수저(귀족)에게 어울리는 분야다. GPU가 없거나 돈없는 분들은 꼭 한번 활용하시기 바란다.

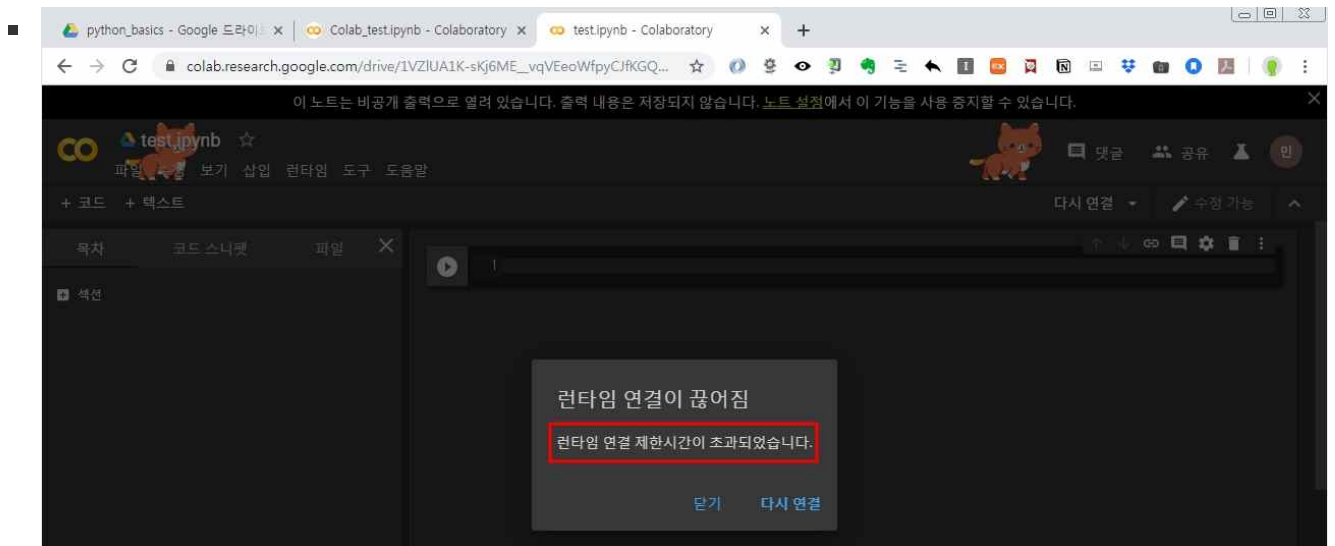
먼저, Colab의 장단점을 알아보자. (참고로, PC 환경은 딥러닝 개발 환경 구축 한방에 끝내기를 참고하시기 바란다.)

- 내 PC도 좋은데.. 굳이 써야되나요?
  - 공짜다.
  - 쉽다. 환경설정 및 구동 준비가 5분이면 끝난다.
  - 클라우드 기반이다.
    - 여러명이 동시에 수정 가능하다.
    - 인터넷 브라우저만 깔려있으면 언제 어디서든 접속하여 수정이 가능하다. 심지어 모바일에서 수정할 수 있다.
  - 어지간한 개인 PC보다 성능이 좋고 빠르다.
    - 비록 내 PC 사양이 뛰어나더라도 딥러닝 학습시간동안 내 PC로 아무것도 못하는것 보다는 영화라도 한편보는 것이 낫지 않은가?
  - 학습 및 공유에 최고!
    - 일반 Jupyter Notebook에 비해 더 좋은 기능을 제공한다. (목차기능, Markdown의 미리보기 기능, 파워레벨, 고양이모드 등)
    - Git과의 연동이 용이하여 타인과 지식을 공유하기 좋은 환경이다.
    - 어딘가 모여 스터디를 진행한다면? Python 기반 특성 상 다들 노트북의 환경 맞추기 등에 시간을 낭비하기 마련인데 그럴 걱정이 없다.
  - 오류발생 시 [SEARCH STACK OVERFLOW] 버튼을 클릭하면 자동으로 스택오버플로우 사이트 검색 결과가 나타난다.

## • 주의할 점

세상의 모든것은 Trade-off가 존재하는 법. 공짜로 쓰는 대신 다음 사항은 유의해야 한다.

- 최대 세션 유지시간은 12시간이다.
  - 아무것도 안하거나 또는 12시간이 지나면 알아서 세션이 끊긴다는 의미이다.



- 세션이 끊기면? 작업중이던 데이터가 다 날라간다.
  - 그럼 쓰면 안되는거 아냐? 소스코드는 .ipynb 확장자로 구글 드라이브에 안전하게 보관되므로 걱정할 필요없다.
  - 다만, 딥러닝 학습시킬 데이터가 문제인데 구글 드라이브에 저장해 놓으면 된다.
    - 물론 개인이 공짜로 쓰는 구글 계정의 최대 용량은 15G이다. 30G이상 저렴한 비용으로 쓸 수도 있다.
    - 학생이라면 구글 GSuite 서비스로 구글 드라이브를 무제한으로 활용하는 것도 방법이다.
- 금융권 등 망분리 보안 이슈로 법적으로 클라우드에 데이터를 올릴수 없는 경우는 사내에서 활용하기 어렵다.

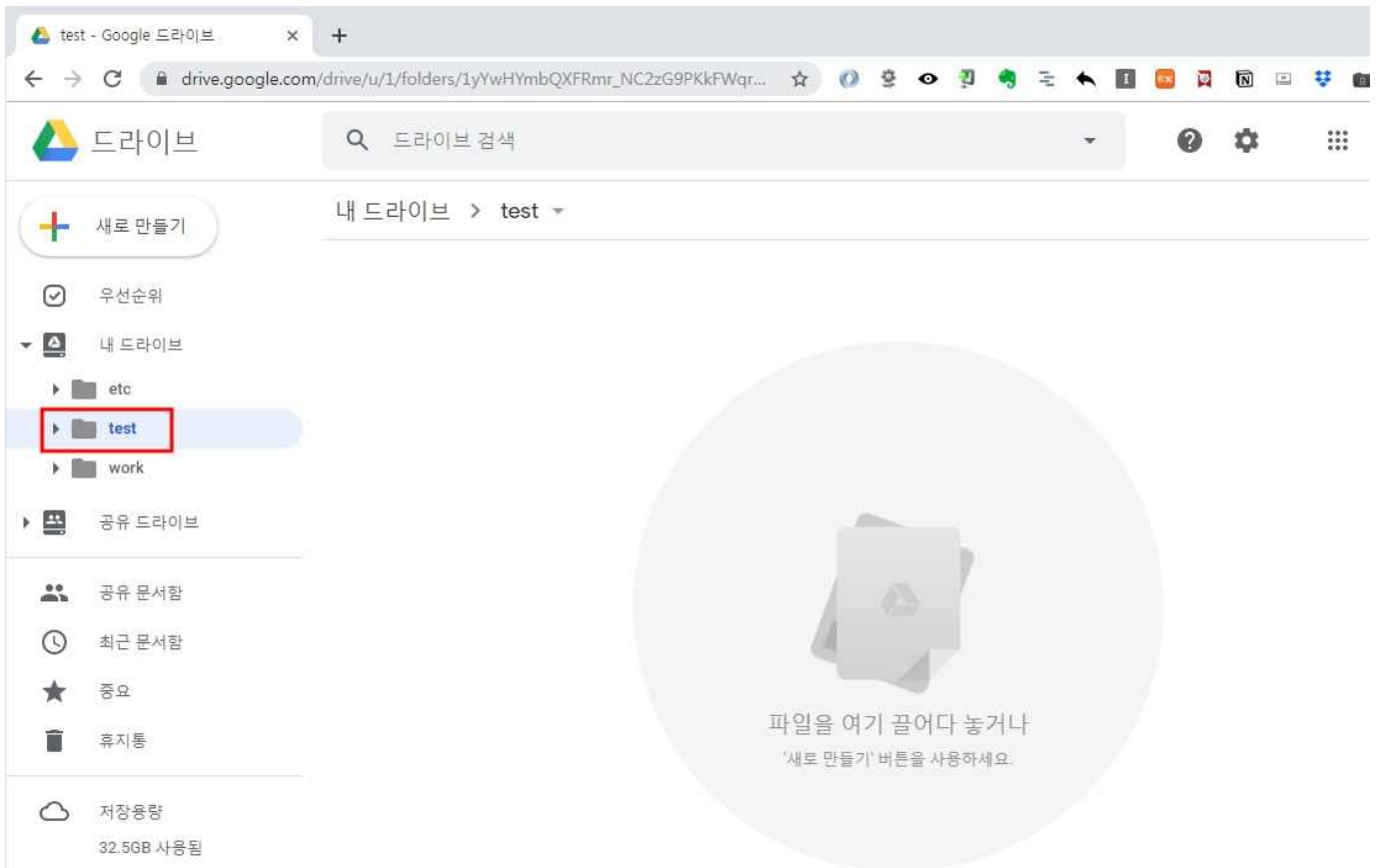
- 보다 자세한 사항은 Google Colaboratory 공식페이지를 참고하시기 바란다.

자! 그럼 이 좋은것을 써 보기 위해 환경설정을 해보자. 위에서 말한대로 5분이면 끝난다.

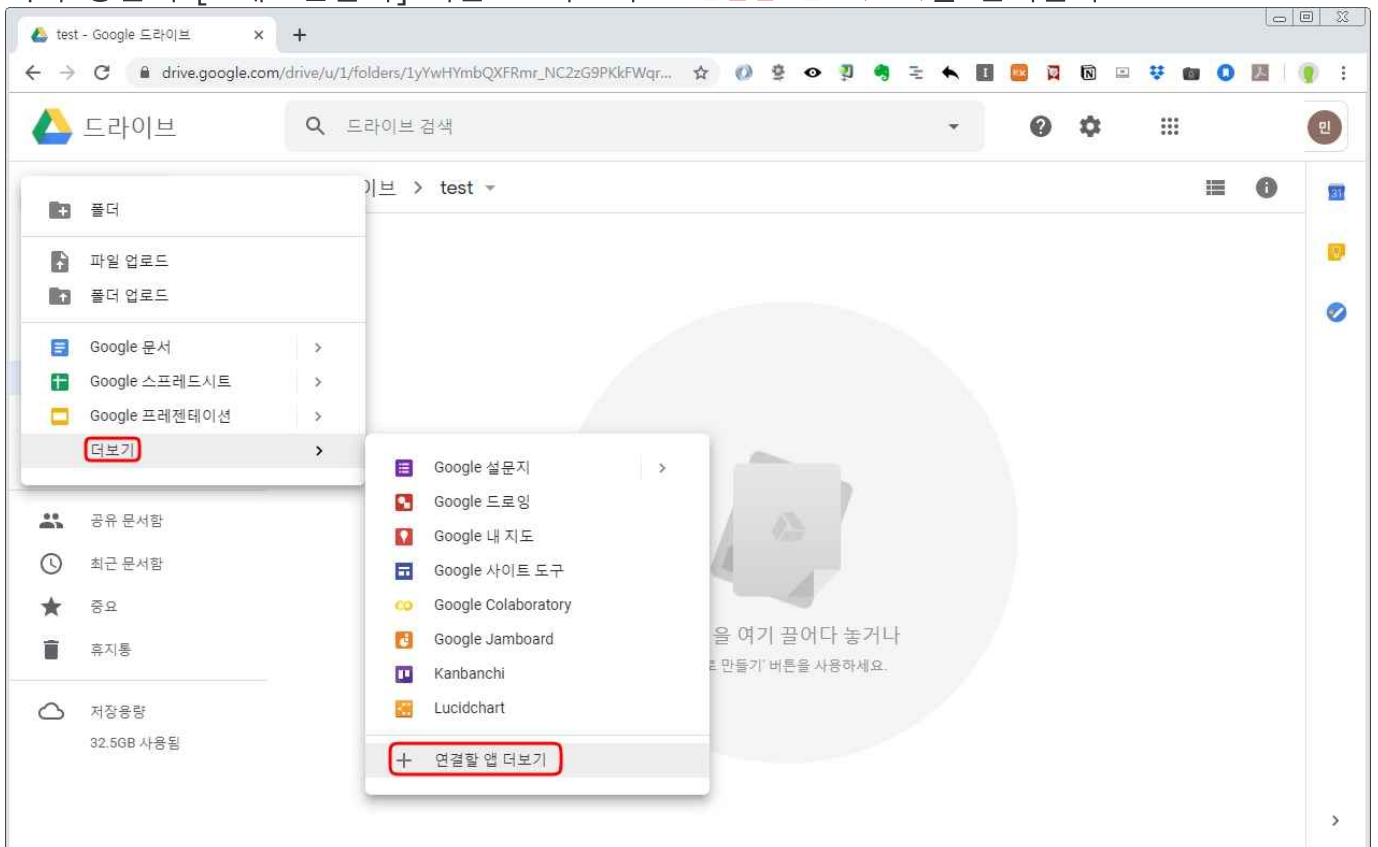
## Colab 환경설정

구글 계정에 가입한 후 아래와 같이 진행한다.

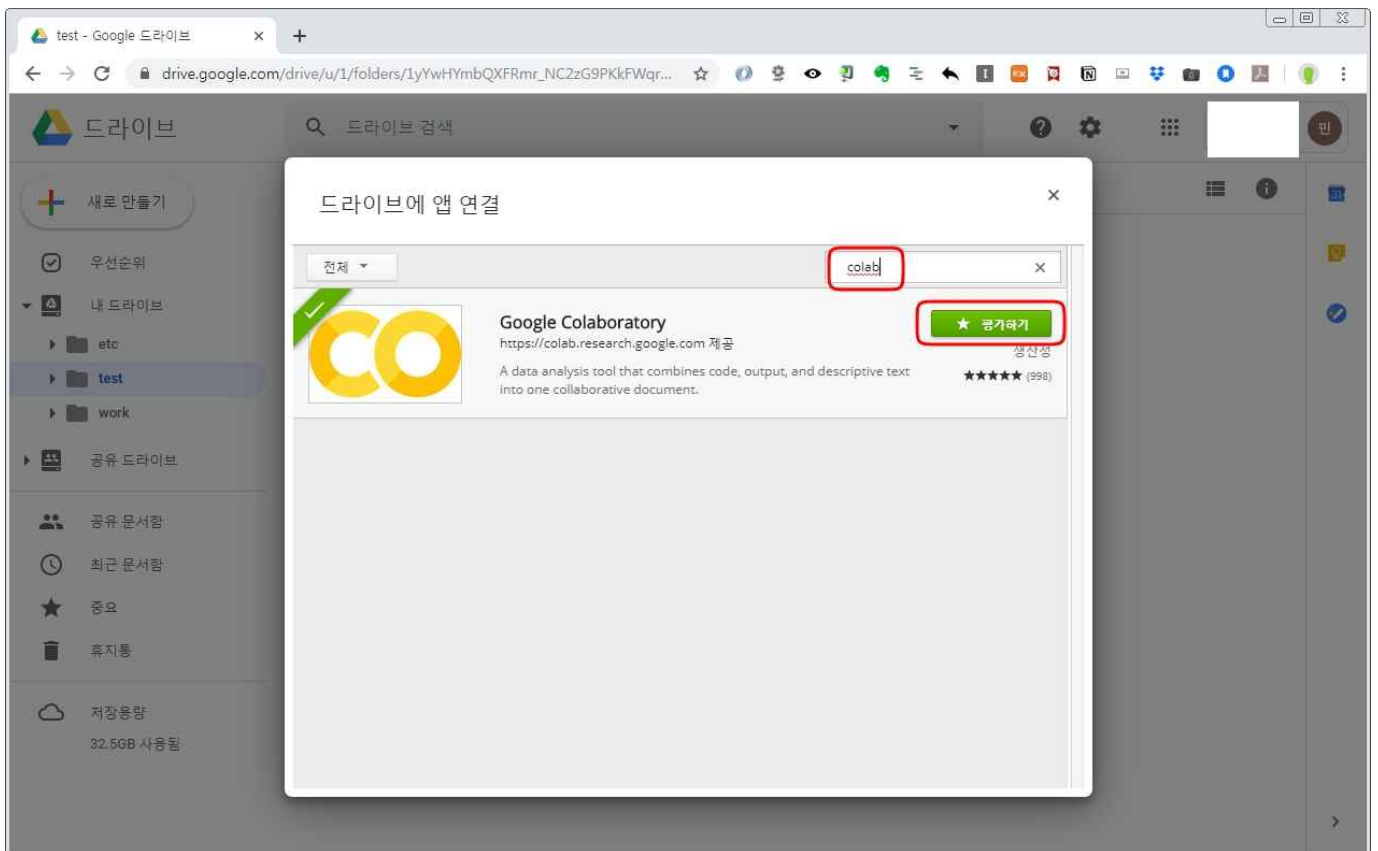
1. <https://drive.google.com>에 접속 후, 우클릭하여 다음과 같이 test 폴더를 만든다.



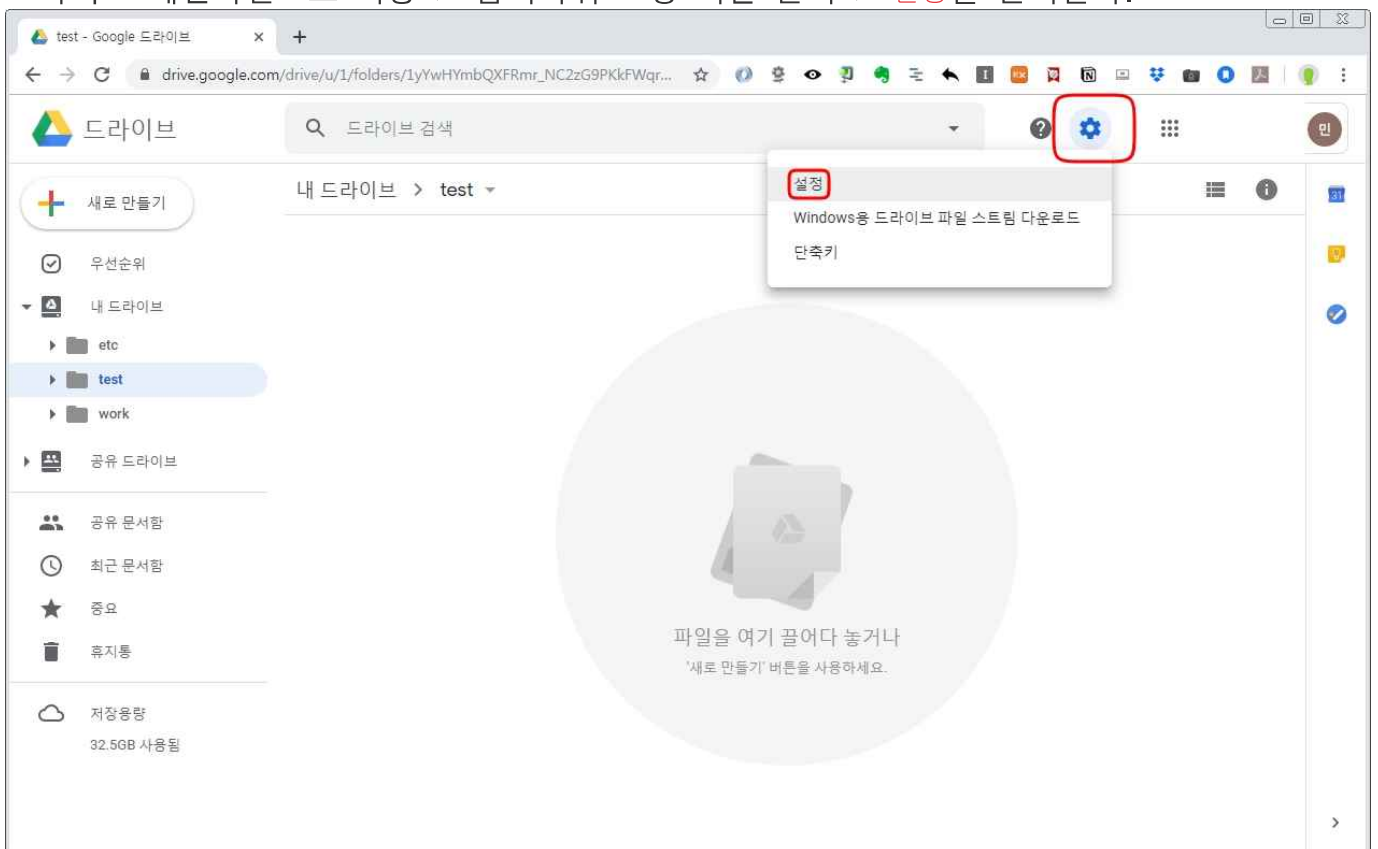
2. 좌측 상단의 [+새로만들기] 버튼 > 더보기 > 연결할 앱 더보기를 선택한다.



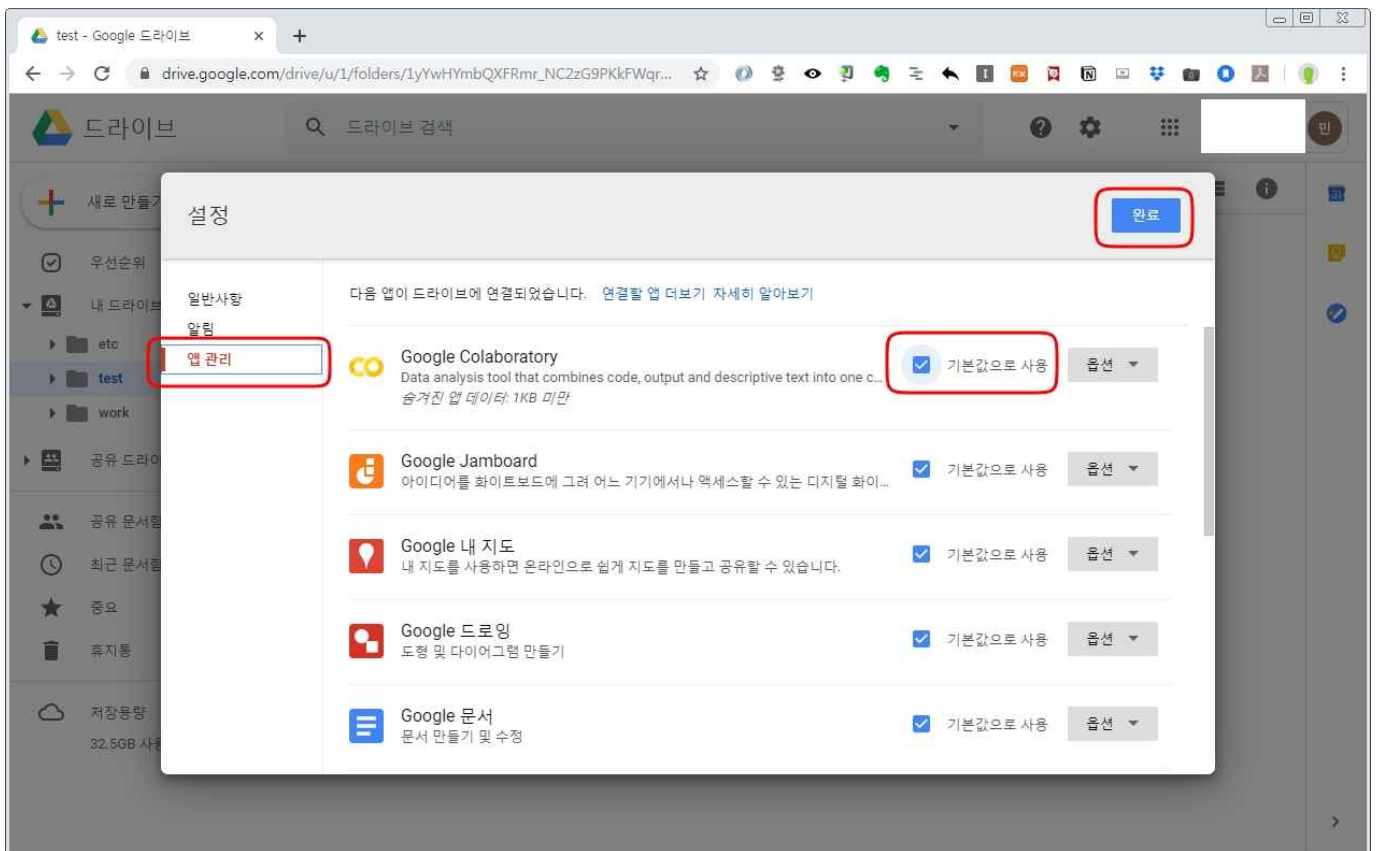
3. 새로 뜬 팝업의 우측 상단에 **colab**이라는 검색어를 입력하면 아래 그림과 같이 Google Colaboratory 앱이 등장한다. 검색어 바로 밑에있는 녹색 **연결하기** 버튼을 누른다. (필자의 경우 이미 연결했기 때문에 평가하기 버튼으로 보이고 있다.)



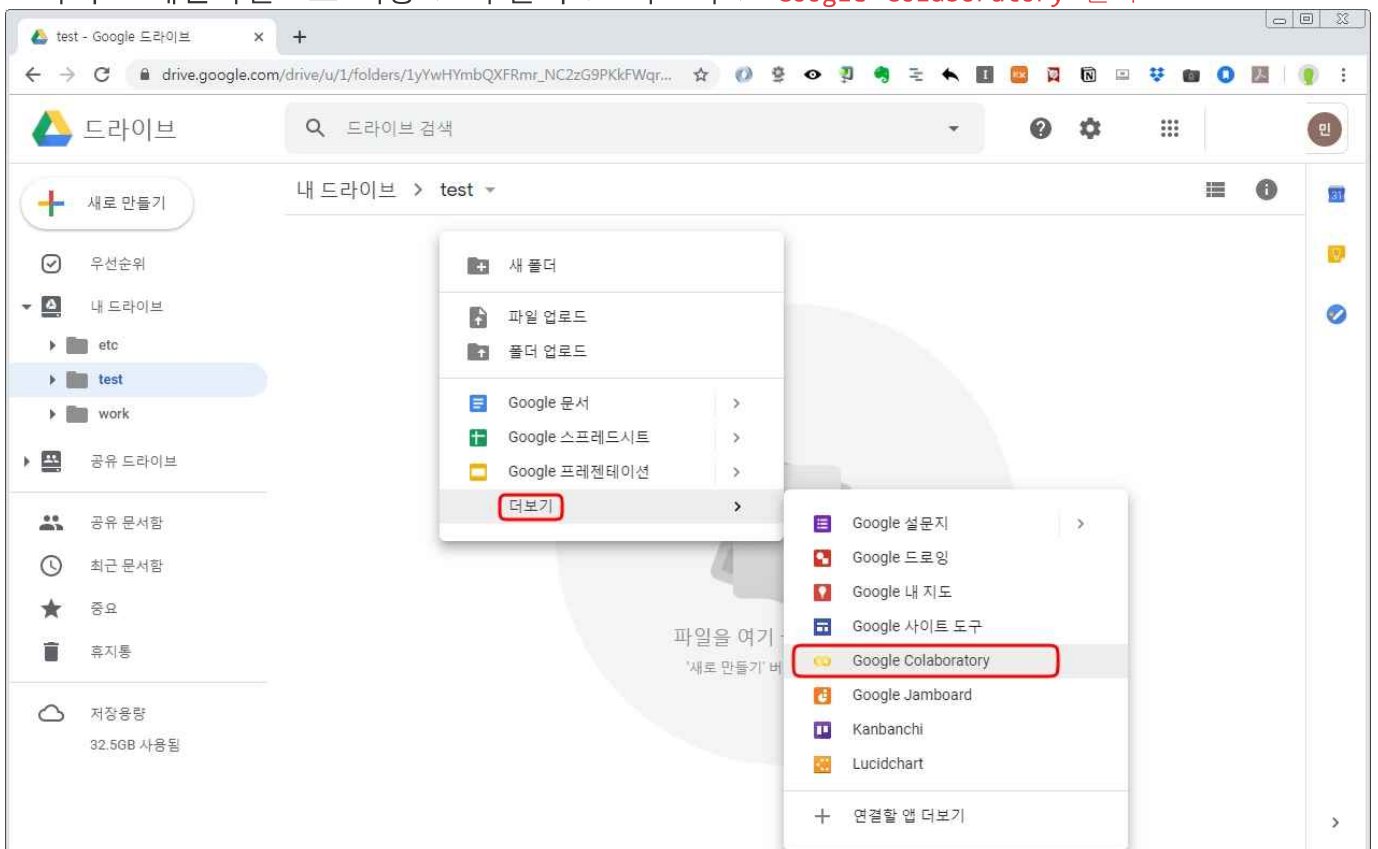
4. 드라이브 메인화면으로 이동 > 톱니바퀴 모양 버튼 클릭 > **설정**을 클릭한다.



5. 새로 뜬 팝업의 좌측 메뉴 앱관리 클릭 > Google Colaboratory 우측의 **기본값으로 사용** 체크박스 클릭 > **완료**를 클릭한다.

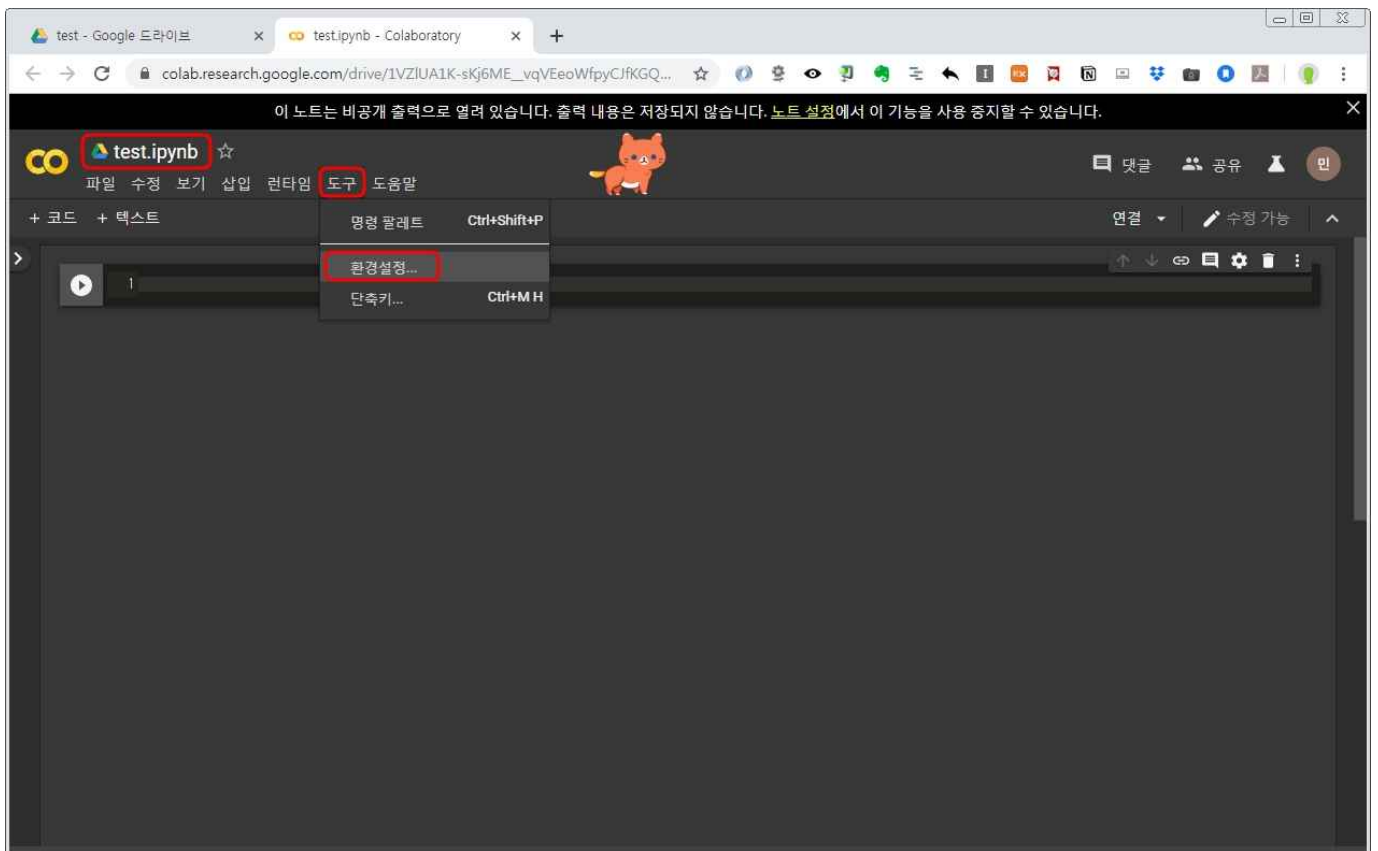


6. 드라이브 메인화면으로 이동 > 우클릭 > 더보기 > Google Colaboratory 클릭

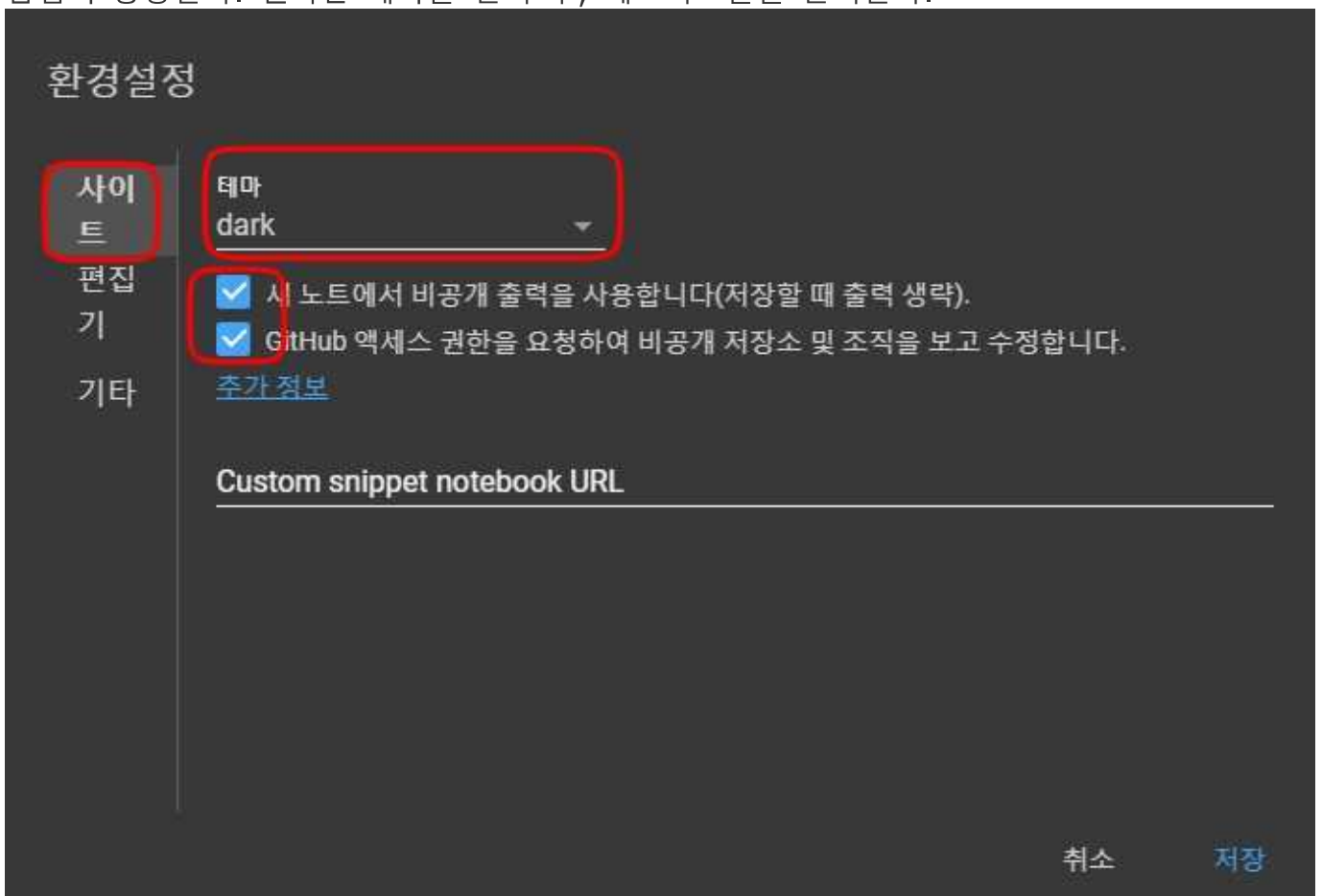


7. 드디어 .ipynb 확장자 파일의 주피터 노트북이 등장했다. 파일이름을 클릭하여 test.ipynb로 이름을 변경하자.

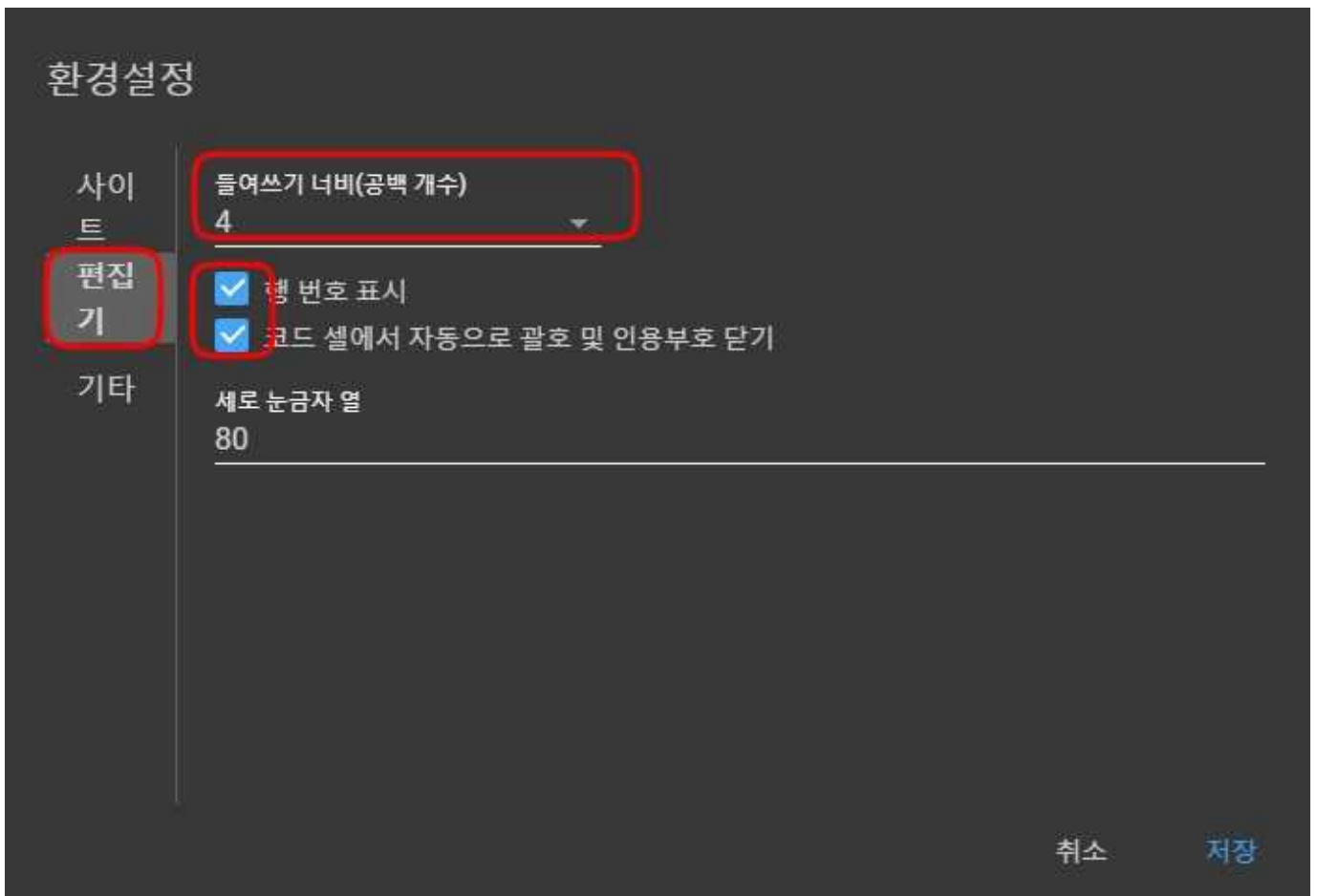
8. 상단 메뉴의 도구 > 환경설정을 클릭한다.



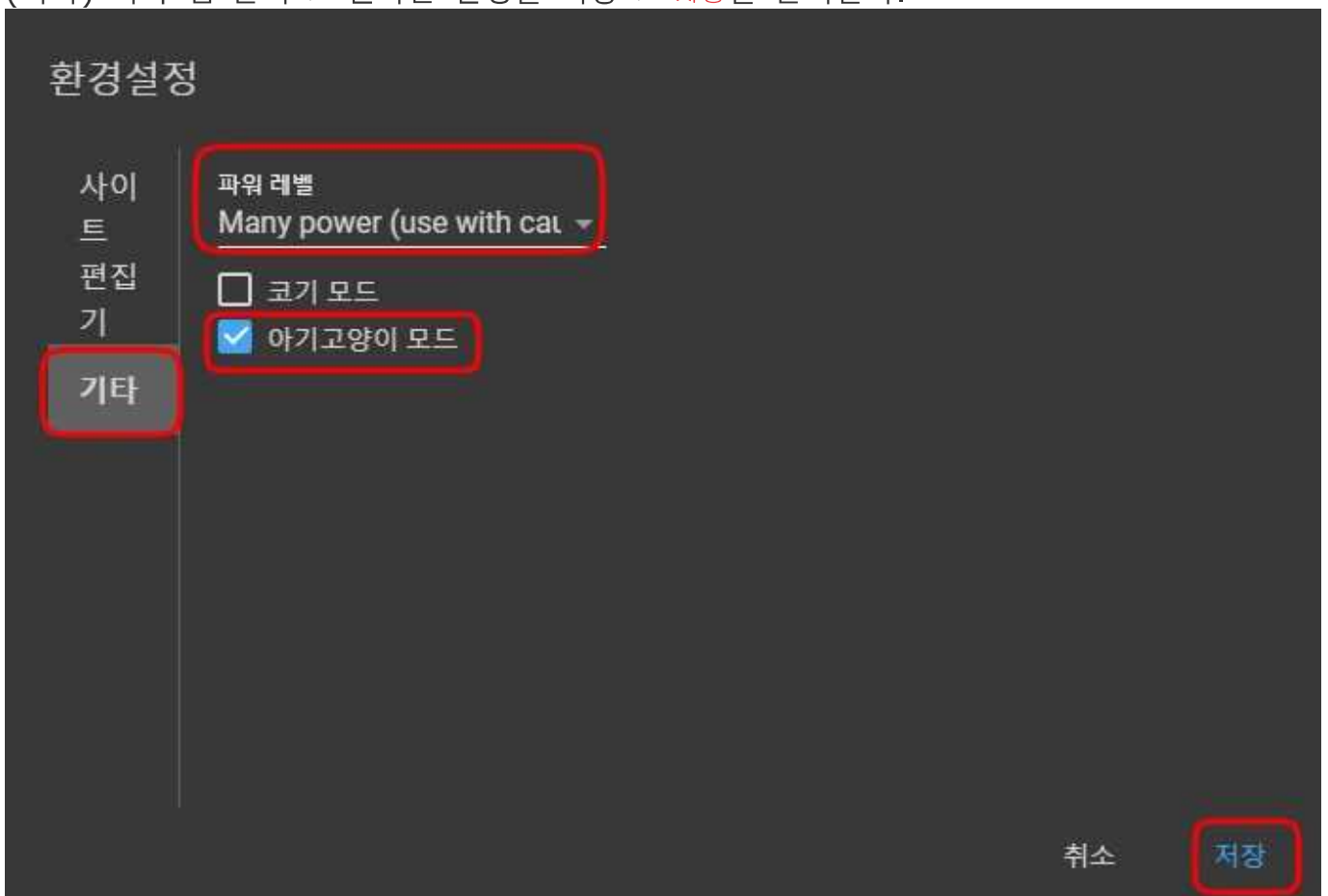
9. 팝업이 등장한다. 원하는 테마를 선택 후, 체크박스들을 클릭한다.



10. (좌측) 편집기 탭 클릭 > 들여쓰기 4선택 > 체크박스 2개 체크



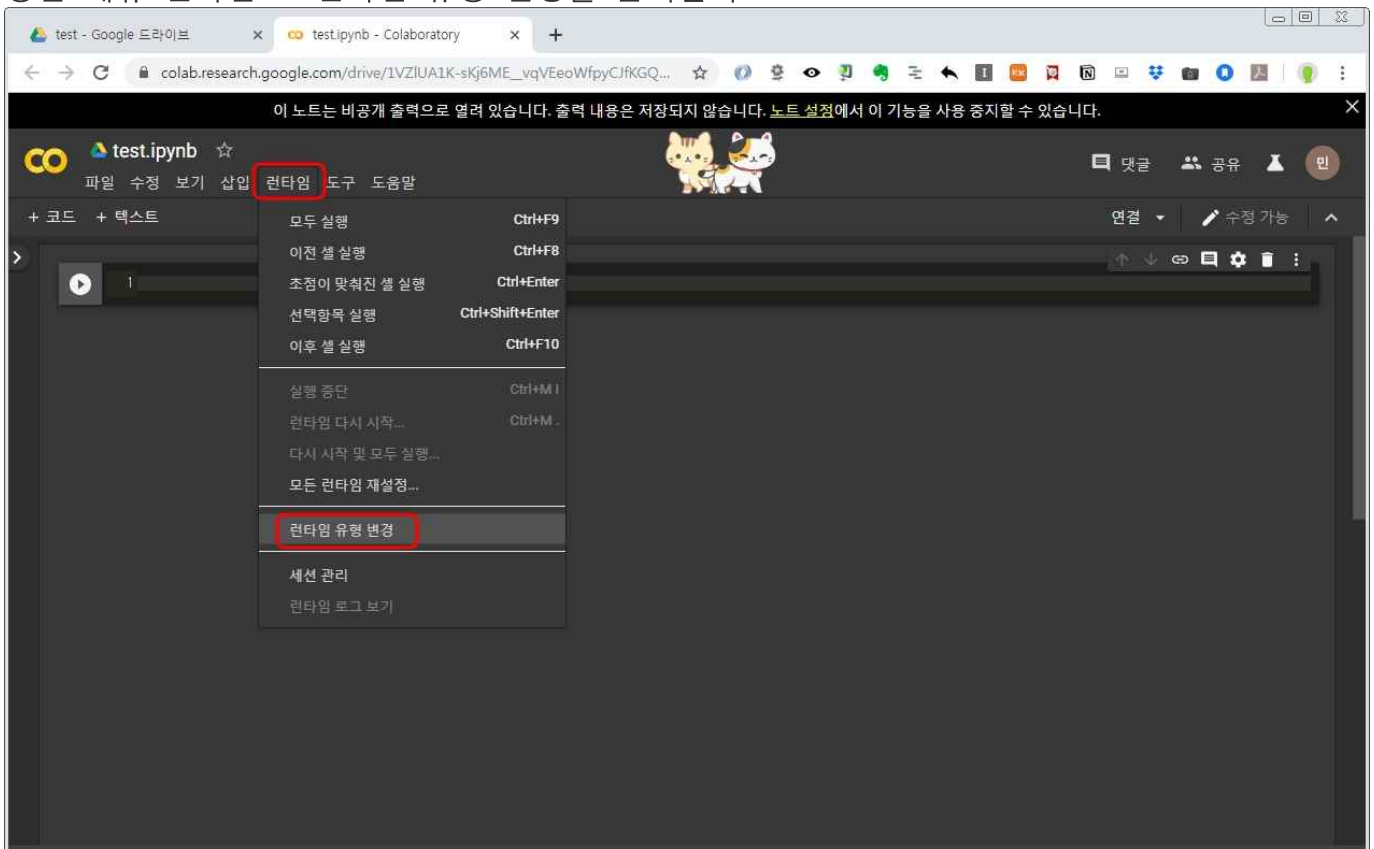
1. (좌측) 기타 탭 클릭 > 원하는 설정을 적용 > 저장을 클릭한다.



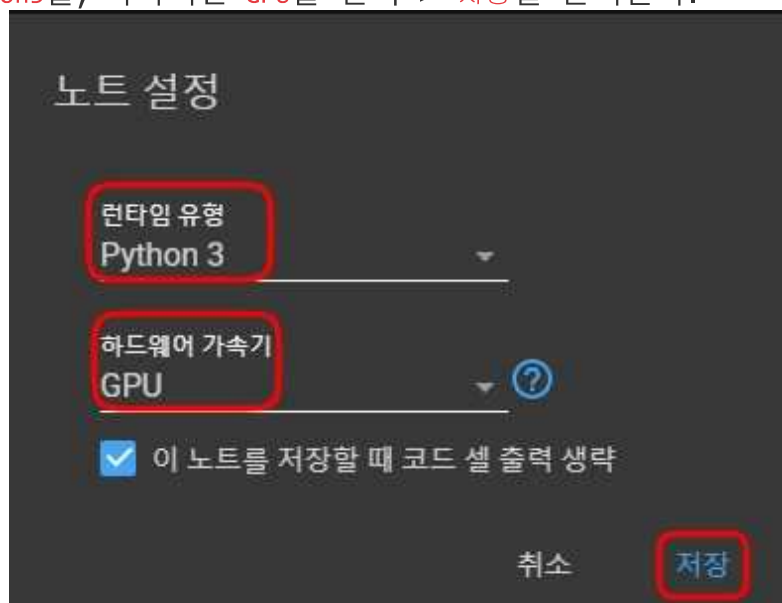
◦ 참고로 파워레벨을 선택 시, 코딩마다 불꽃이 튀기는 재미가 있다.

- 아기고양이 모드를 선택 시, 코딩에 지칠 때 고양이들이 위에 튀어나와 근심(?)을 덜어준다.

l2. 상단 메뉴 런타임 > 런타임 유형 변경을 클릭한다.

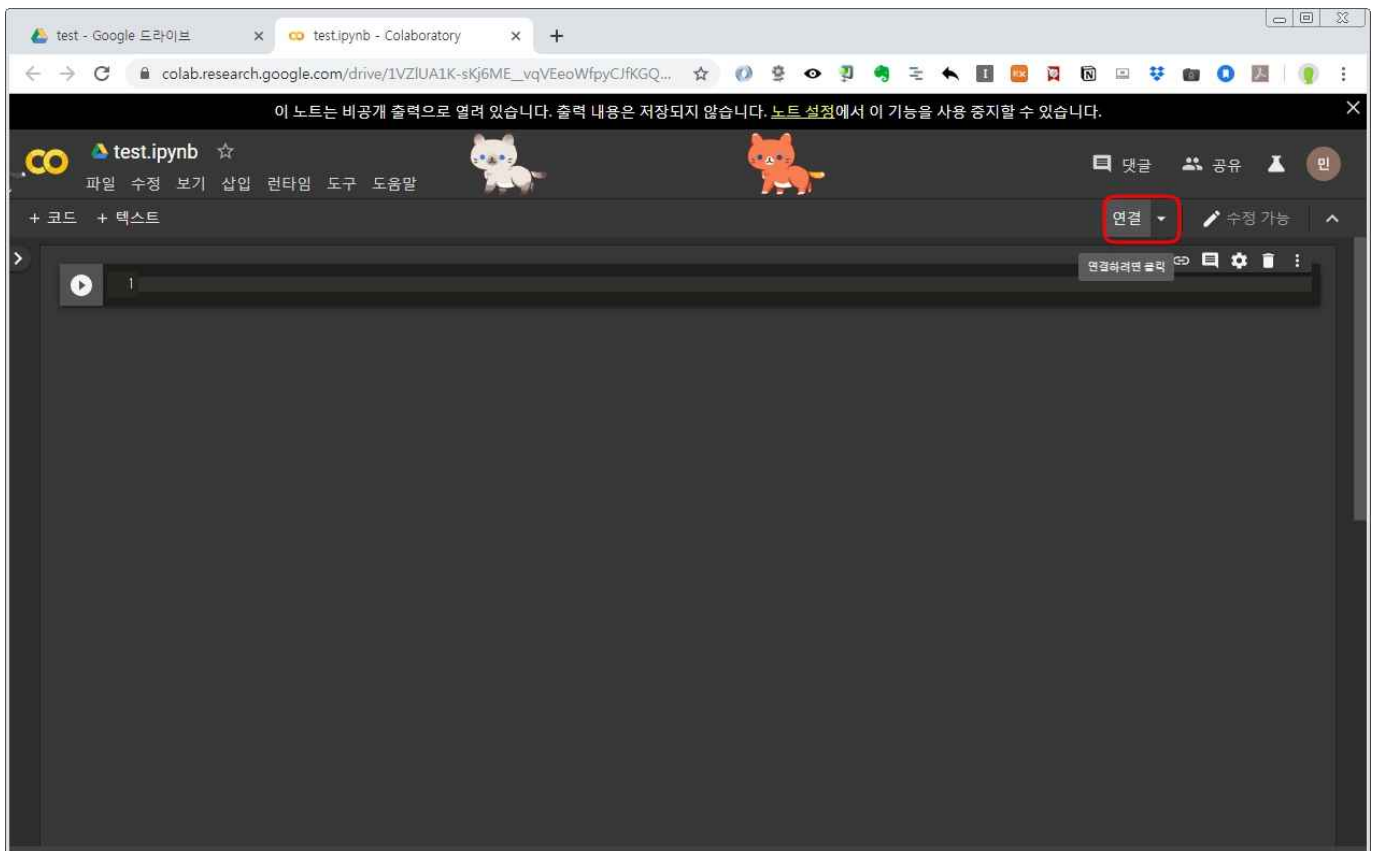


l3. 런타임 유형은 Python3를, 가속기는 GPU를 선택 > 저장을 클릭한다.



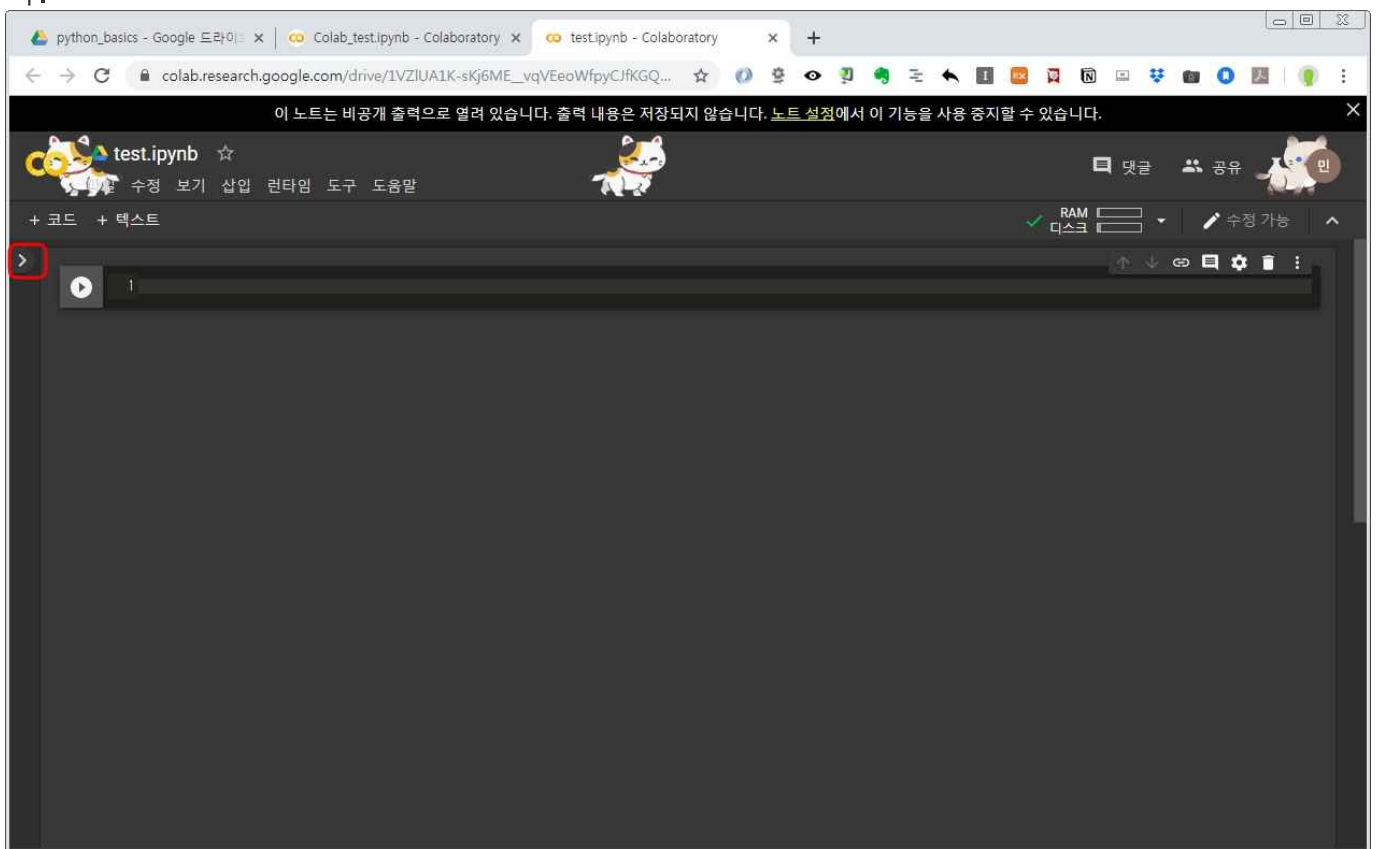
l4. 아래 그림과 같이 연결 버튼을 클릭한다.

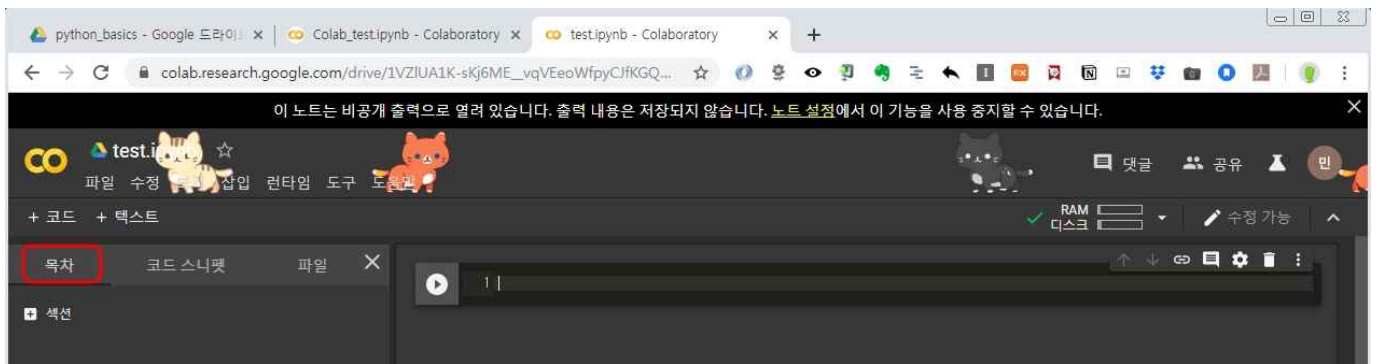




◦ 할당중.. > 연결중... > 초기화중.. 으로 텍스트가 변경되며, 최종 RAM, 디스크 사용량 막대 그래프가 나올 것이다.

15. 아래와 같이 [>] 모양의 버튼을 클릭하면, 목차, 코드스니펫, 파일 등의 기능을 활용할 수 있다.



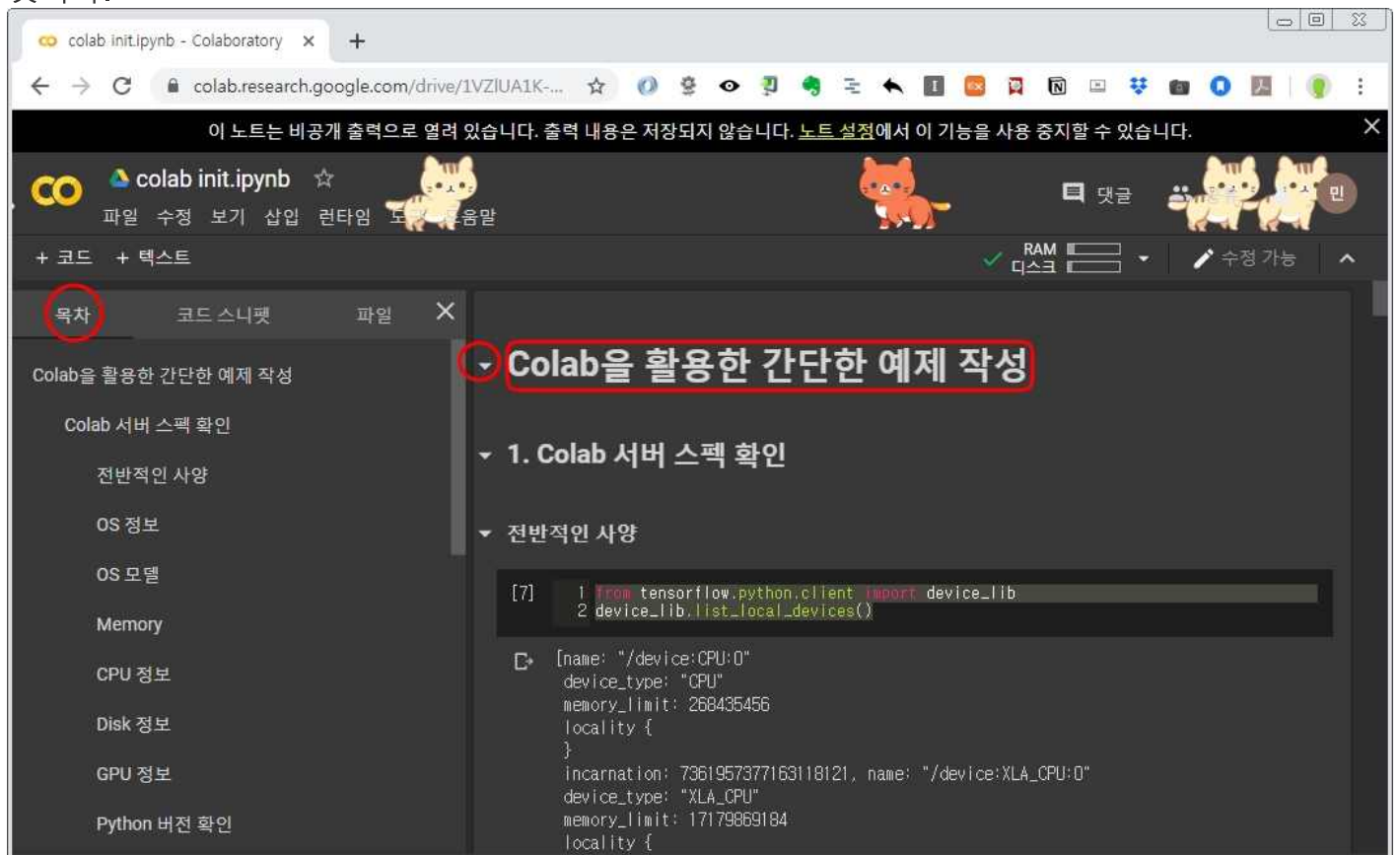


- 주피터 노트북 수천줄 코드 속에서 셀끼리 비교를 하느라 마우스 드래그에 지치신 적이 있으시다면 할렐루야!가 저절로 나올것이다.
- 위에서 설명했듯이 Markdown의 기능 또한 강력하다. 다음 다음 챕터에서 설명하겠다.

자! 이것으로 Colab 환경설정은 끝났다. 너무 쉽지 않은가? 이제 간단한 예제를 작성해보자.

## Colab을 활용한 간단한 예제 작성

먼저 필자의 Colab .ipynb 파일 링크를 클릭하여 접속해주시기 바란다. 아래와 같은 화면이 보일 것이다.



이 파일은 Colab에서 필수적으로 사용하게 될 클라우드 원격서버 스펙 확인, 파일다루기, 구글 드라이브 연동, 텐서플로우 및 케라스의 예제를 필자가 직접 테스트해 본 Colab 실습예제이므로 복사 또는 공유를 통해 반드시 따라하며 실습하시기 바란다. 10분 정도만 투자하면 전체 매커니즘을 파악하는데 어렵지 않게 될 것이다. 각 셀을 실행하는 방법은 **Ctrl+Enter** 단축키를 입력하시면 된다.

목차를 활용하셔서 참고하시면 편리하며, 그림에 빨간동그라미로 가리킨 [▼]모양의 버튼을 클릭하시면 다른 문단의 셀이 자동 접기가 되므로 해당 Chapter에만 집중하여 보실 수 있다.

아래에는 블로그 포스트 특성 상 간단한 예제 코드를 기술하였으며, 자세한 내용은 상단의 링크를 참고하시기 바란다.

### \* Colab 서버 스펙 확인

```
from tensorflow.python.client import device_lib
device_lib.list_local_devices()
```

```
import platform
platform.platform()
```

아래의 코드는 한줄씩 셀에서 실행하기 바란다. 쥬피터 노트북의 셀은 출력결과가 여러개인 경우 맨 마지막의 `OutputStream` 결과가 남으므로 이전 출력결과를 확인할 수 없기 때문이다. 일반적인 `.py`파일 작성과는 다르게 `.ipynb`파일의 경우 가급적 하나의 셀에 많은 코드를 작성하지 않는것이 좋다.

```
!cat /etc/issue.net
!cat /proc/meminfo
!cat /proc/cpuinfo
!df -h
!nvidia-smi
!python --version
!ls
```

```
from tensorflow.python.client import device_lib
device_lib.list_local_devices()
```

### \* 파일처리

```
%writefile test.py
print('hello world!')
```

```
# test.py 실행시키기
%run test.py
```

```
from google.colab import files
# 브라우저에 다운로드 됨을 확인할 수 있다.
files.download('test.py')
```

```
# [Cancel upload] 버튼을 클릭하여 잠시 멈춘 후 파일선택 버튼을 클릭하면 PC 내 파일을 선택할 수
# 리턴값을 받는 변수인 myupload라는 이름의 디렉토리가 생성된다.
myupload = files.upload()
```

## \*구글드라이브 연동

```
import os
print(os.getcwd())
!ls

# 실행시 등장하는 URL을 클릭하여 허용해주면 인증KEY가 나타난다. 복사하여 URL아래 빈칸에 붙여넣으면
from google.colab import drive
drive.mount('./MyDrive')

# 마운트된 내 드라이브를 확인해보자
!ls

# 해당 드라이브로 이동
# 내 드라이브는 원격서버가 아니라 로컬서버로 간주하므로 명령어 실행시 앞단에 !를 붙이지 않는다.
cd MyDrive/My Drive

# 내드라이브의 전체 목록이 나타난다.
ls

# 특정파일을 가져오고 싶은 경우 다음과 같이 접근한다.
import pandas as pd
df = pd.read_csv("./MyDrive/test/test.csv")
```

텐서플로우 및 케라스 예제 실행은 코드가 긴 관계로 생략하며 상기 링크를 통해 확인하시기 바란다.

## Colab & Markdown

예전 포스팅에서 이미 마크다운(Markdown) 사용법 및 예제를 자세히 소개한 바 있다. 이 장에서는 당시 다루지 않은 특히 Colab에 특화된 새로운 마크다운 작성방법 및 단축키에 대하여 설명하겠다.

하단의 단축키는 매우 자주 사용되는 단축키이므로 반드시 숙지하시기 바란다. 마우스로 이동하며 일일이 버튼을 클릭하는 것은 노가다가 심해 불편하다.

- 선택된 셀을 실행 : Ctrl + Enter
- 선택된 셀을 실행 후 다음 셀로 포커스 이동 : Shift + Enter
- 실행 후 다음줄로 이동 : Alt + Enter
- 선택모드에서 화살표 방향키 : 셀 포커스를 위 아래로 움직일 수 있음
- 엔터키 : 편집모드(Vi 편집기와 유사)
- ESC : 선택모드(Vi 편집기와 유사)
- 마크다운으로 전환 : Ctrl + M M
- 코드로 전환 : Ctrl + M Y
- 저장 : Ctrl + S

- 코드셀에 줄번호 부여 : Ctrl + M L
- 바로 윗줄에 셀 생성 : Ctrl + M A
- 바로 아랫줄에 셀 생성 : Ctrl + M B
- 셀 삭제 : Ctrl + M D
- 셀 병합 : (shift를 누른 상태에서 병합을 원하는 셀들을 한번에 다중 선택 후), Shift + M
- 셀 분할 : (분기를 원하는 부분에 커서를 지정 후), Ctrl + Shift + -
- 코드가 오래 실행되어 멈추고 싶은경우 : Ctrl+ M + I
- 위 코드로도 멈추지 않고 작업을 완전종료하고 싶은 경우 : Ctrl+M+.