

기초 컴퓨터 프로그래밍

- Python 개요 및 개발환경 설치 -

충남대학교 의과대학 의공학교실

구 윤 서 교수

Python 개요

- 개발자: Guido van Rossum (1956년 1월 31일~)
- 네덜란드 출신의 컴퓨터 프로그래머
- 2005~2012년까지 구글 근무
- 2013년부터 드롭박스에서 근무



Python 개요

- “1989년 12월,
저는 크리스마스 주중에
저의 "취미"가 될만한 프로
그램을 찾고 있었습니다.”

(2008년 구글 개발자 컨퍼런스)

- 1991년 Python 공개



Python 개요

- 1999년, “Programming for Everybody” 에서 Python 목표 정의
 - 당연히 무료
 - 오픈 소스로 누구나 개발 가능
 - 평이한 영어로 이해할 수 있는 코드
 - Web 개발부터 Application 개발, 인공지능 알고리즘 개발까지 다양한 업무에 대한 적합성
 - 짧은 개발 시간

Python 특징

- 운영체제(OS) 독립적인 인터프리터 언어
 - 소스 코드를 바로 실행할 수 있게 지원(Interpreter)하여 윈도우, 맥, 리눅스, 안드로이드 등 프로그램이 실행되는 운영체제에 상관없이 한번 프로그램을 작성하면 사용가능
- 높은 가독성
 - 코드 블록을 들여쓰기로 구분하여 간결하며, 읽기 편함
- 풍부한 라이브러리
 - 매우 광범위한 라이브러리가 기본적으로 포함되어있으며 외부 라이브러리에 대한 확장이 용이

Python 특징

- 객체 지향적 (Object-Oriented)

- 코드를 실행 순서가 아닌 단위 모듈(객체) 우선으로 프로그램을 작성하고 실행 순서와 흐름을 설계
- 하나의 객체는 어떤 목적을 달성하기 위한 데이터 (attribute)와 행동(method)을 가지고 있음.

- 접착성

- 쉽게 라이브러리를 추가할 수 있으며, Python에서 C로 구현된 부분을 사용하거나 C에서 Python 을 사용 가능

Python 설치 (Local)

- **Windows OS** 기준으로 진행
- **Python 3.9** version (2022.9.1 기준)
- **Anaconda**: 데이터 분석에 필요한 오픈소스 Python 개발 플랫폼 (by Continuum Analytics, Inc.)
 - <https://www.codecademy.com/articles/install-python-data-analysis>
- **PyCharm**: IDE(Integrated development environment)
 - <https://www.jetbrains.com/pycharm/>

Python 설치 - Anaconda

- 컨티넘(Continuum)사가 제작한 Python 배포판
- 배포판은 사용자가 설치를 한번에 수행할 수 있도록 필요한 라이브러리들을 하나로 정리해둔 것임.
- 아나콘다 배포판은 데이터 분석에 중점을 둔 배포판임.
- <https://www.anaconda.com/download/> 로 이동 후 환경에 따라서 선택

Python 설치 - Anaconda



Anaconda Repository

Our repository features over 8,000 open-source data science and machine learning packages, Anaconda-built and compiled for all major operating systems and architectures.

Python 설치 - Anaconda

Conda

Conda is an open-source package and environment management system that runs on Windows, macOS, and Linux. Conda quickly installs, runs, and updates packages and their dependencies. It also easily creates, saves, loads, and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

```
[bash-3.2$ conda install pandas
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /opt/anaconda3

added / updated specs:
  - pandas

The following packages will be downloaded:

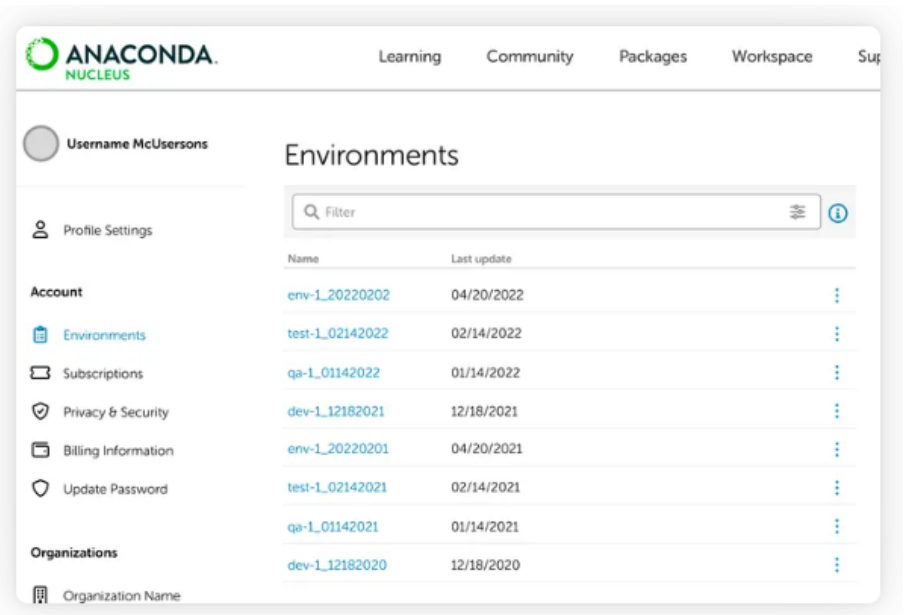
package                        | build                | 9.4 MB
-----|-----
pandas-1.4.1                   | py39he9d5cce_1      |
-----|-----
                                Total:      9.4 MB

The following packages will be UPDATED:
```

Python 설치 - Anaconda

Cloud Environment Backup

Connect Anaconda Navigator to our community portal, [Anaconda Nucleus](#), to securely store your local environments in the cloud.



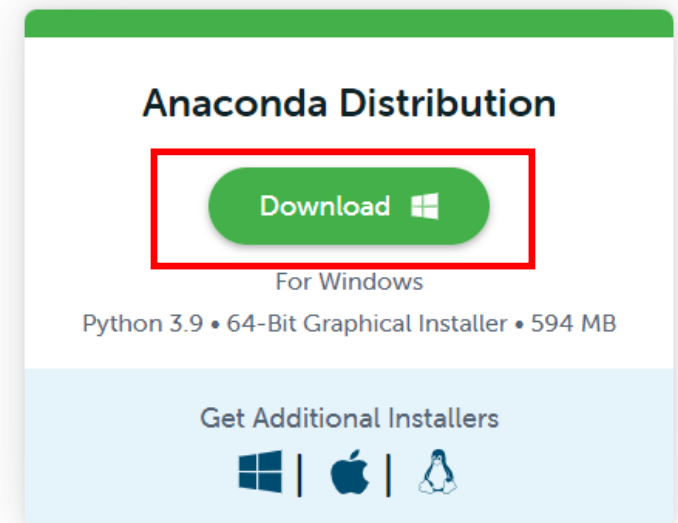
Python 설치 - Anaconda

<https://www.anaconda.com/download/>

Individual Edition is now

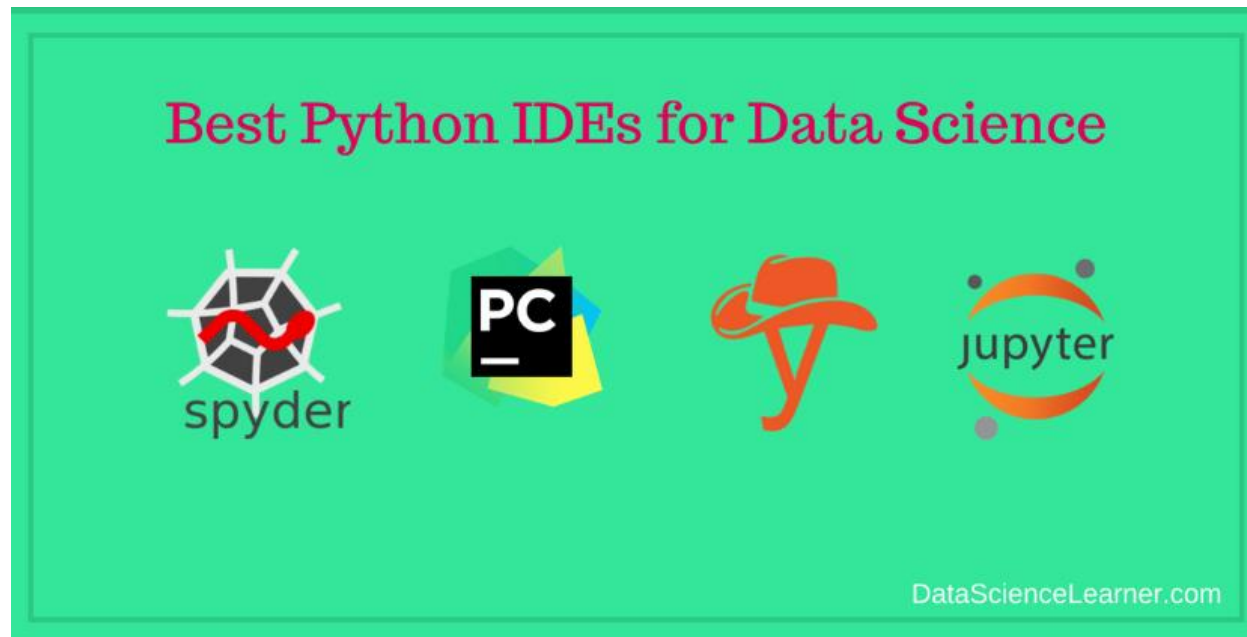
ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform



PyCharm 설치

- 다수의 Python 통합 개발 환경(Integrated Development Environment, IDE)이 존재



PyCharm 설치

- PyCharm: a Popular & Free IDE tool for Python developers



<https://www.jetbrains.com/pycharm/>

PyCharm 설치



Version: 2019.2.2

Build: 192.6603.34

September 11, 2019

[System requirements](#)

[Installation Instructions](#)

[Other versions](#)

Download PyCharm

Windows

macOS

Linux

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

DOWNLOAD

Free trial

Community

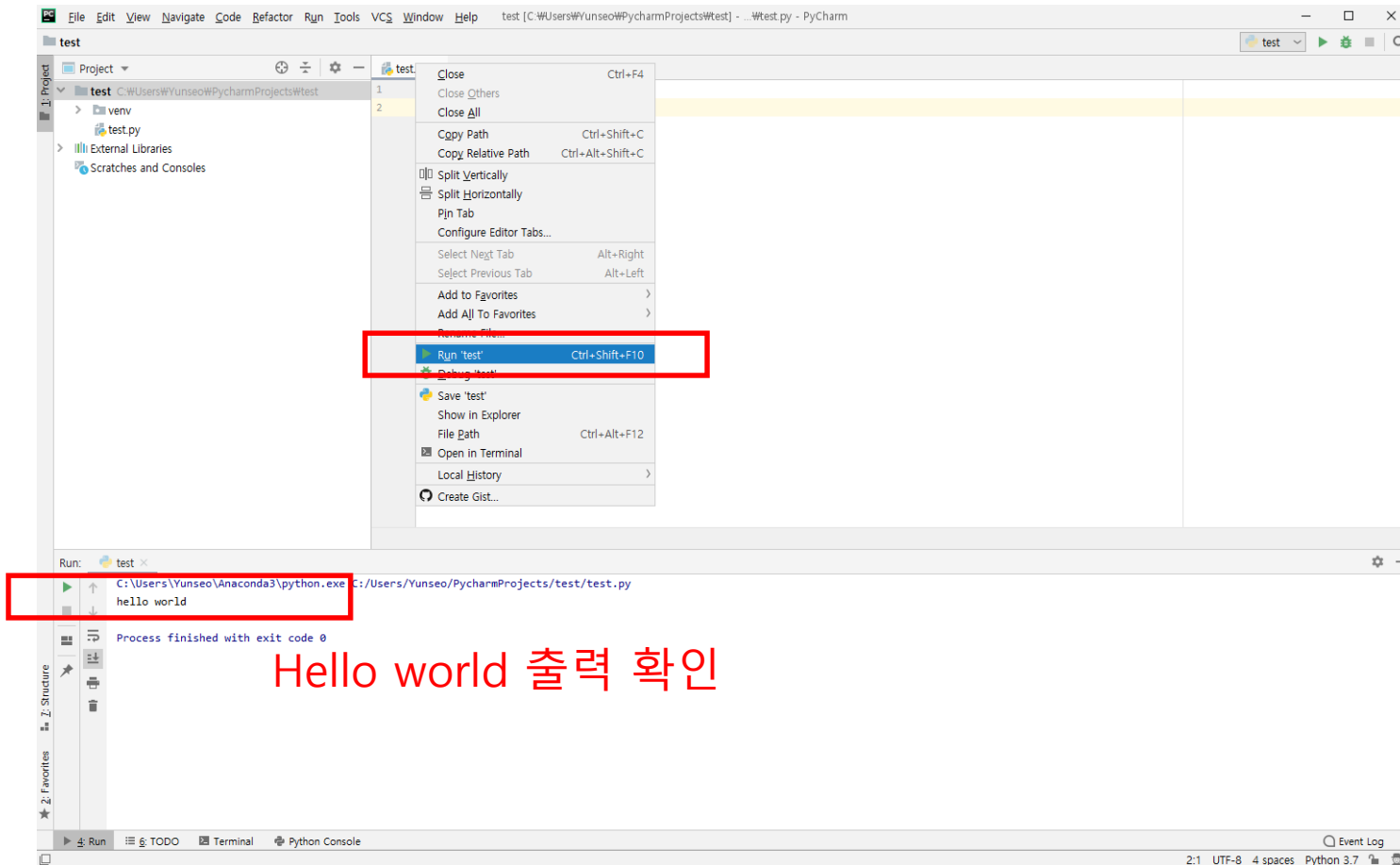
For pure Python development

DOWNLOAD

Free, open-source

Code 실행

- Code 이름에서 우클릭하여 진행해도 실행 가능



Google Colab(oratory)



- Colaboratory is a free [Jupyter notebook](#) environment that requires no setup and runs entirely in the cloud.
- Free GPU support
- Upload/Download files (Google drive)
 - Gmail 계정 필요
- Github Friendly
- Easy Collaboration

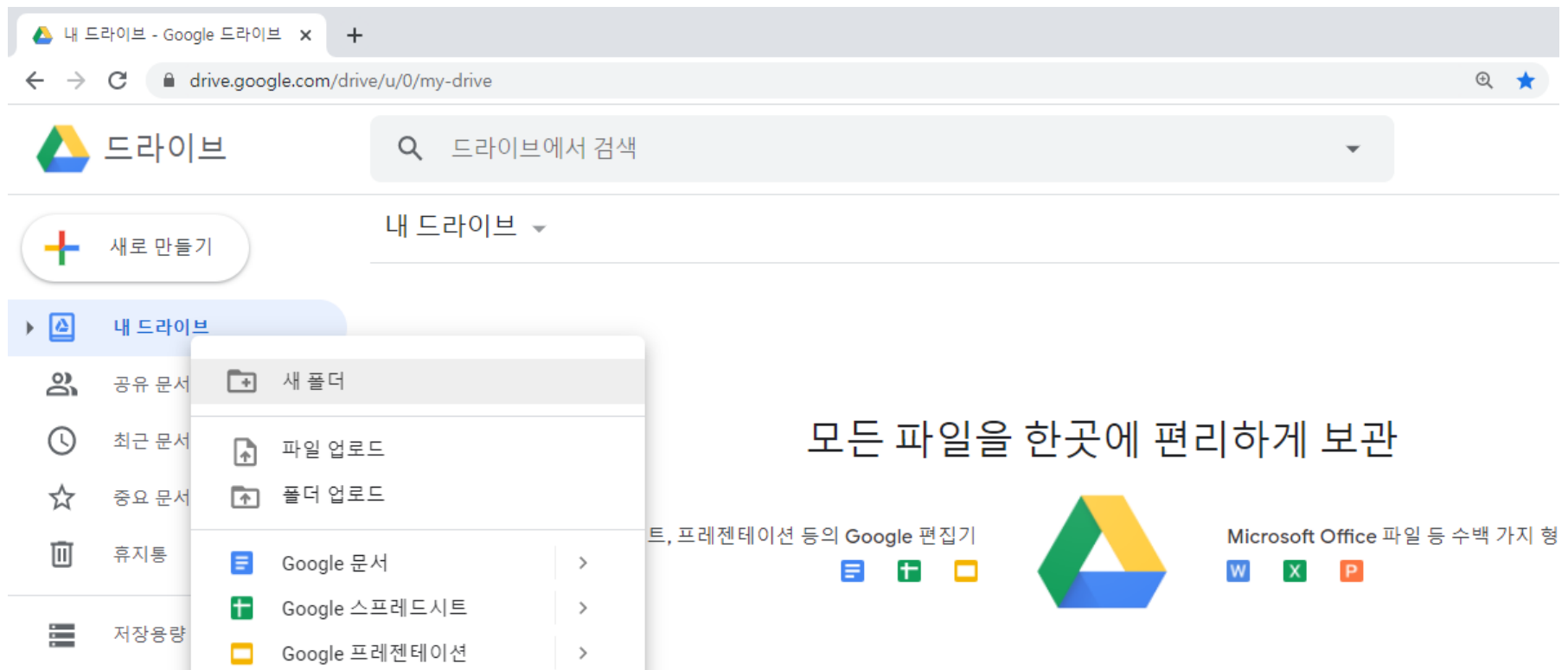
Colab 환경설정

- <https://drive.google.com>에 접속 (my drive)



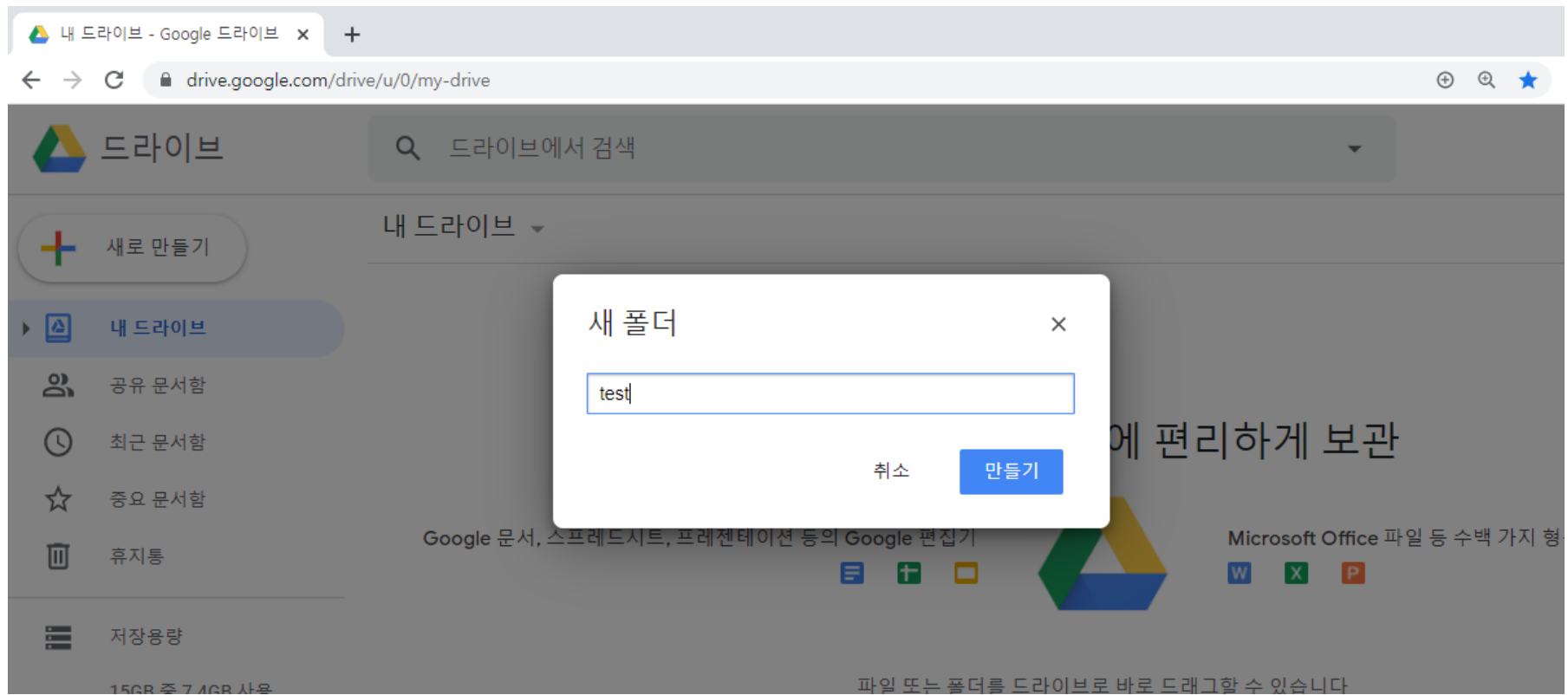
Colab 환경설정

- 우클릭 → 새폴더



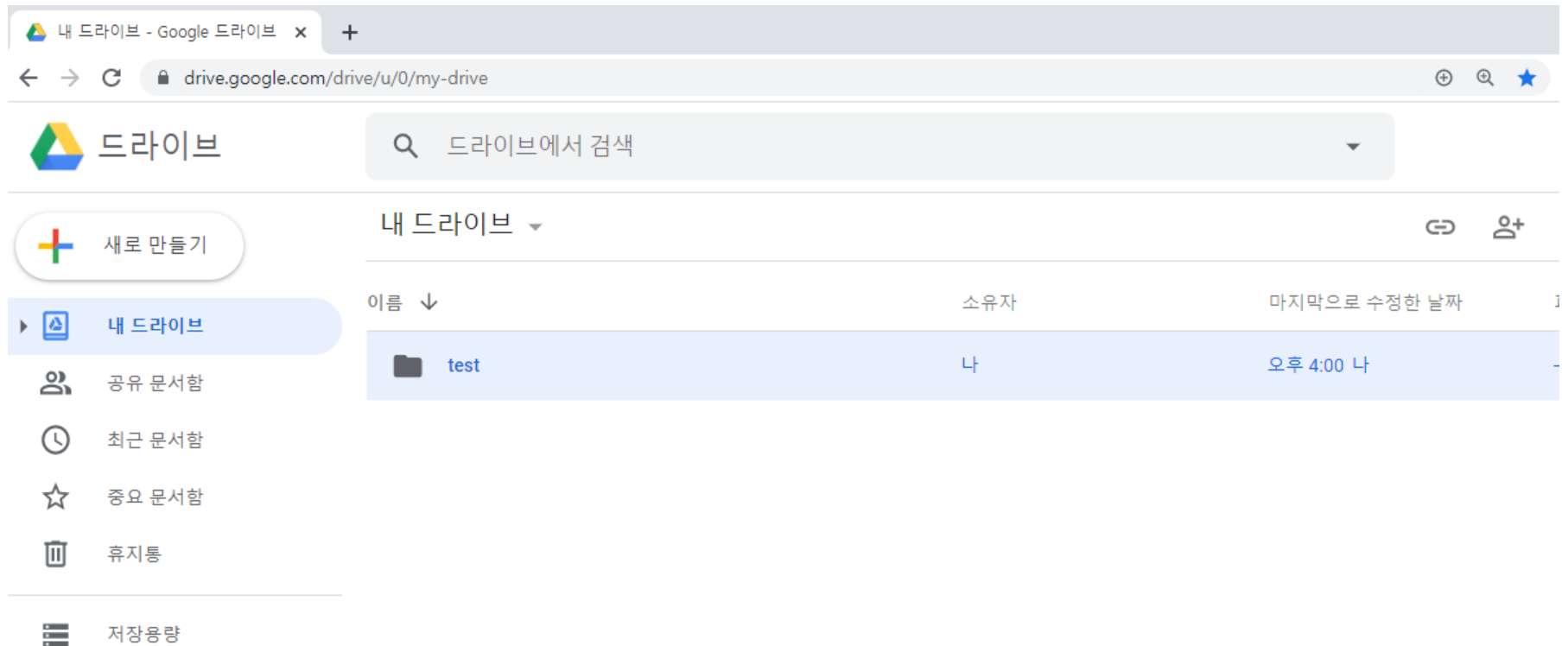
Colab 환경설정

- test 폴더 이름 입력



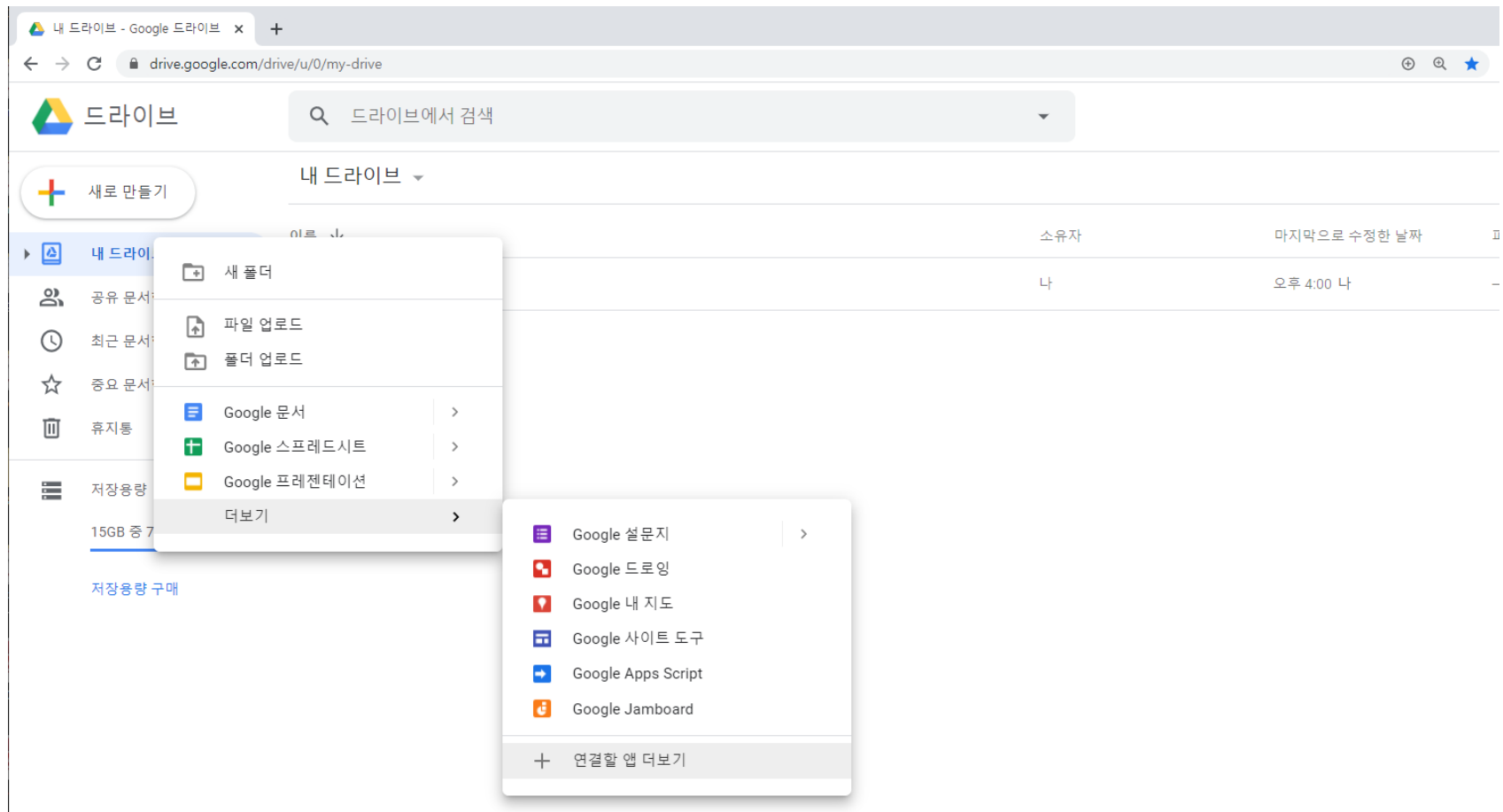
Colab 환경설정

- test 폴더 생성



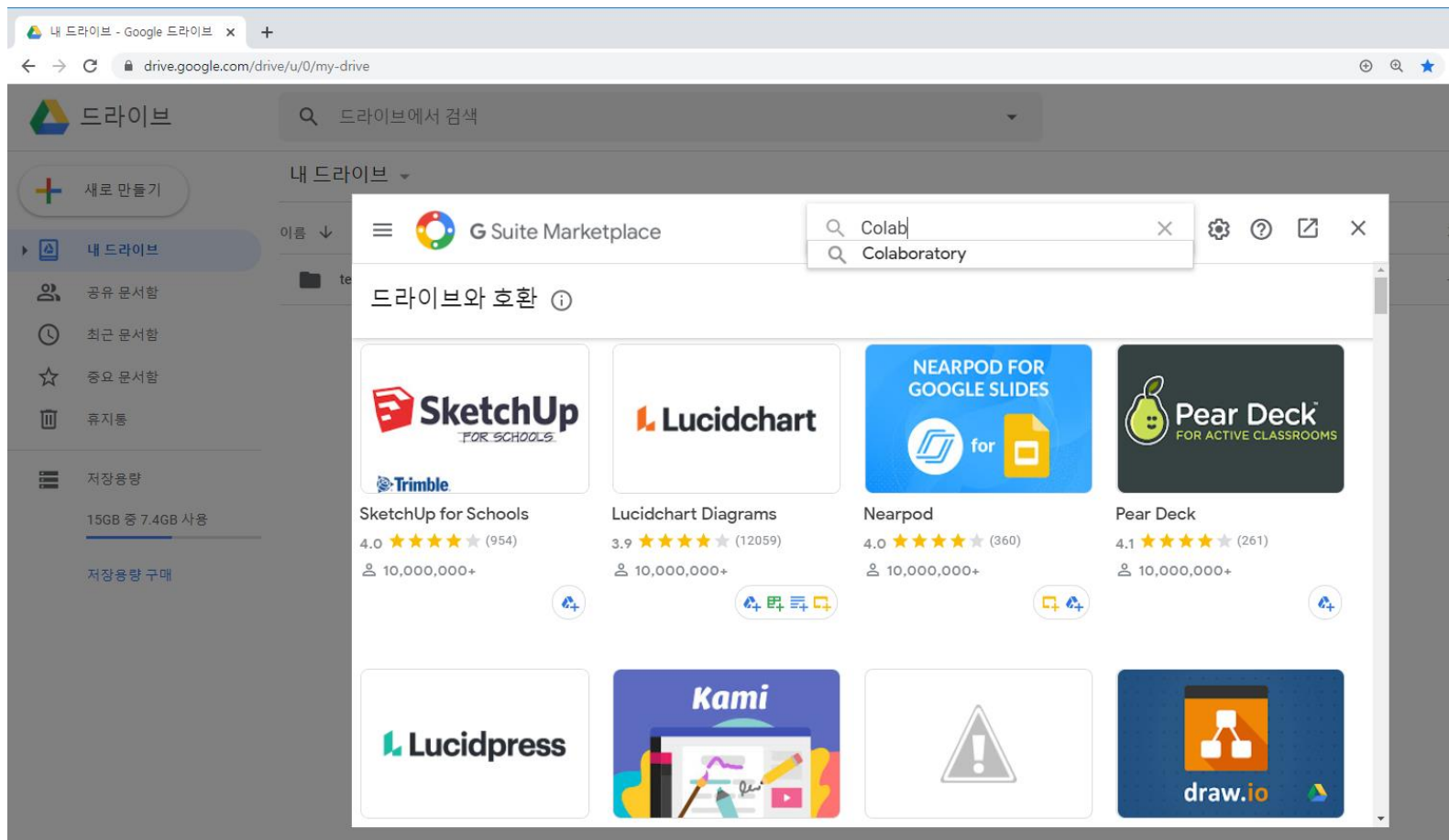
Colab 환경설정

- 우클릭 → 더보기 → 연결할 앱 더보기



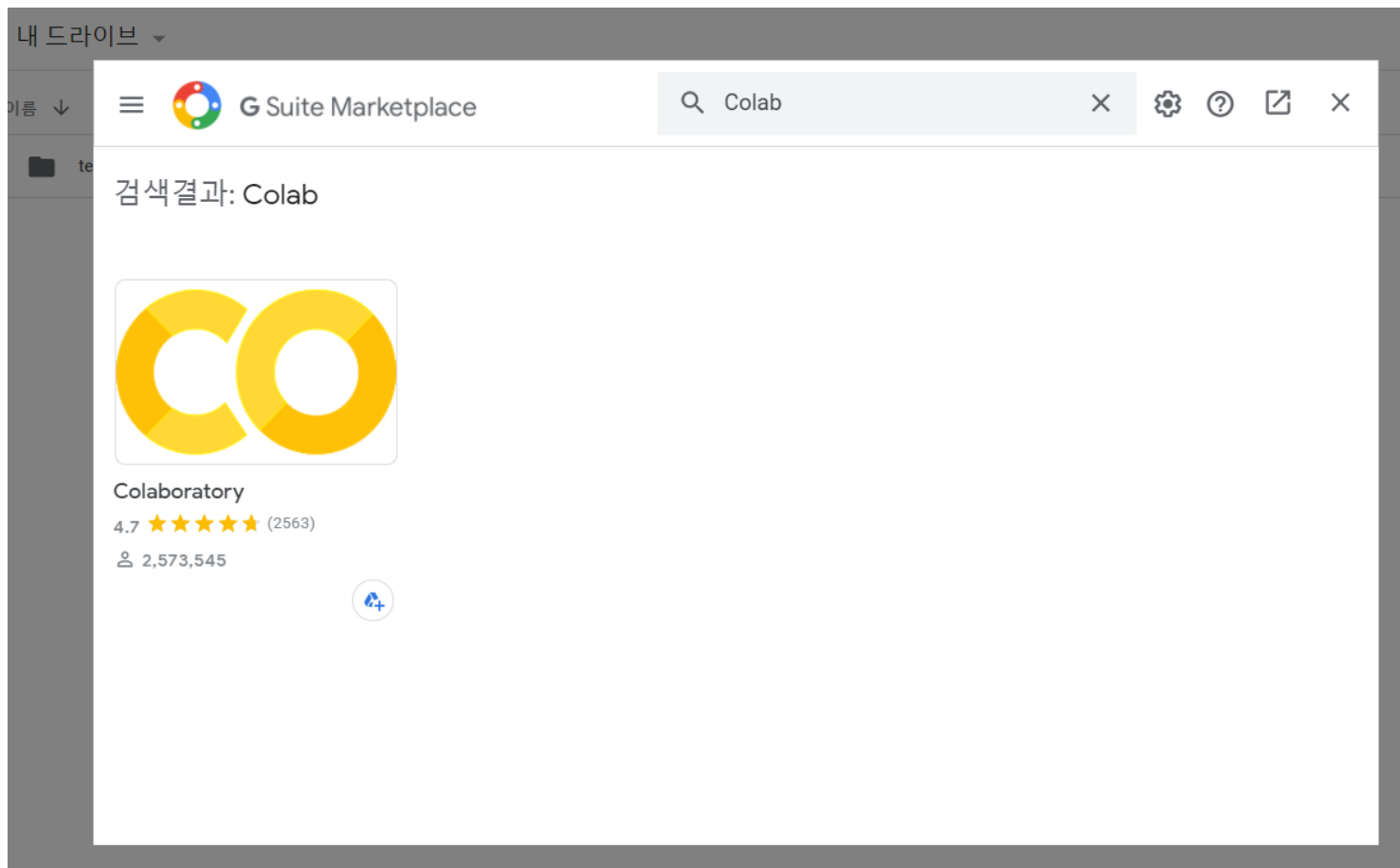
Colab 환경설정

- 검색창에 "colab" 입력



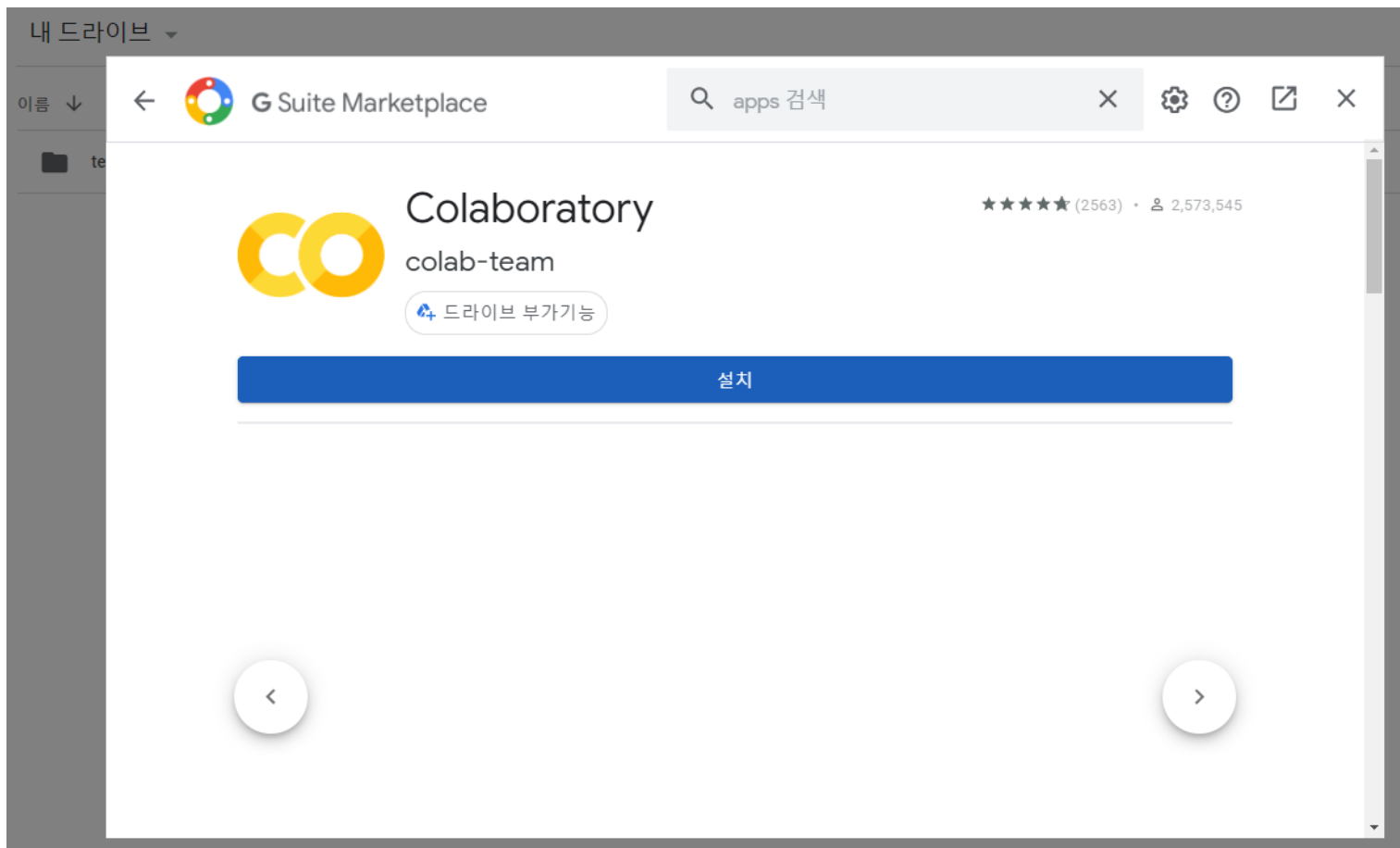
Colab 환경설정

- 검색창에 "colab" 입력



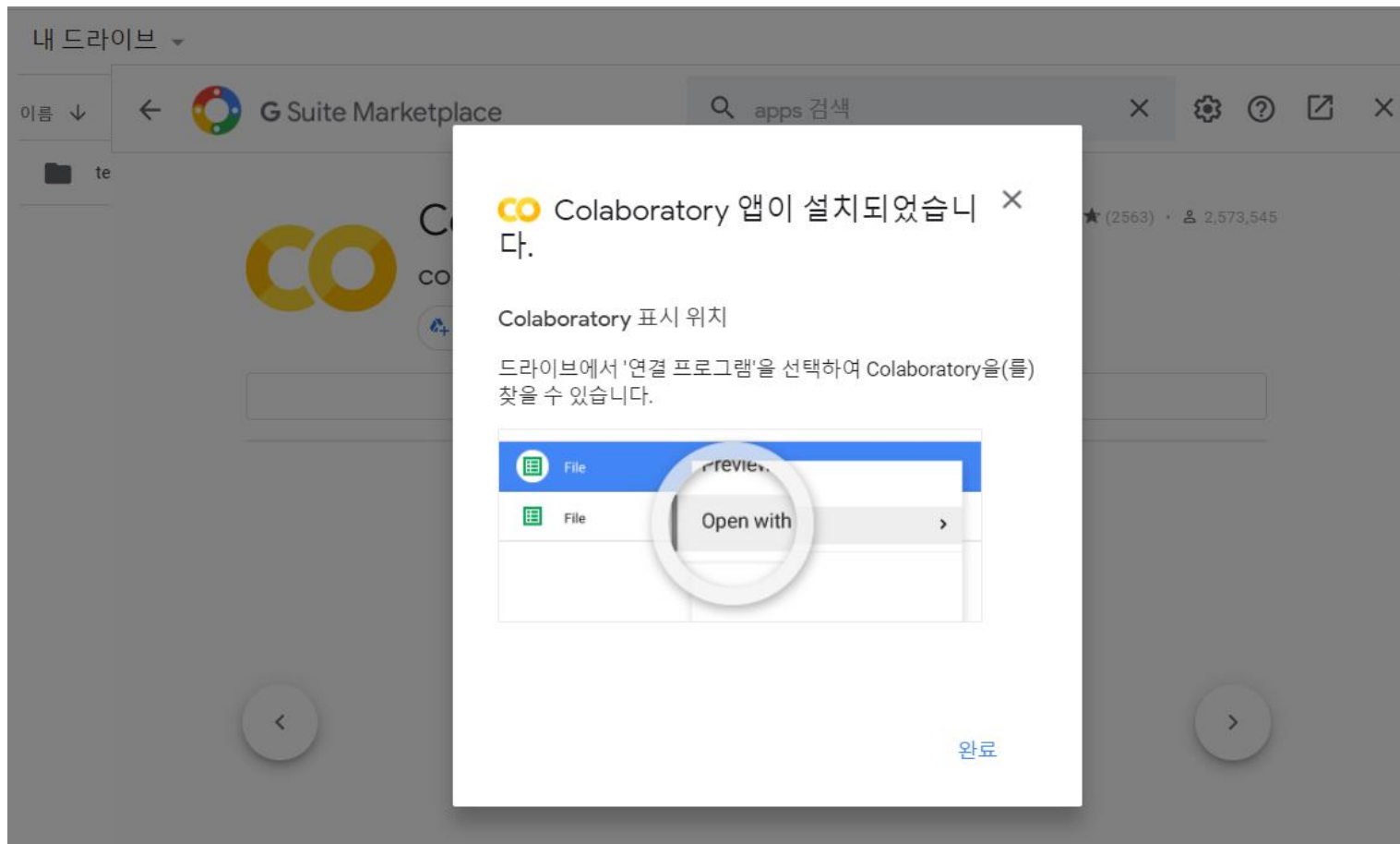
Colab 환경설정

- Colaboratory 설치



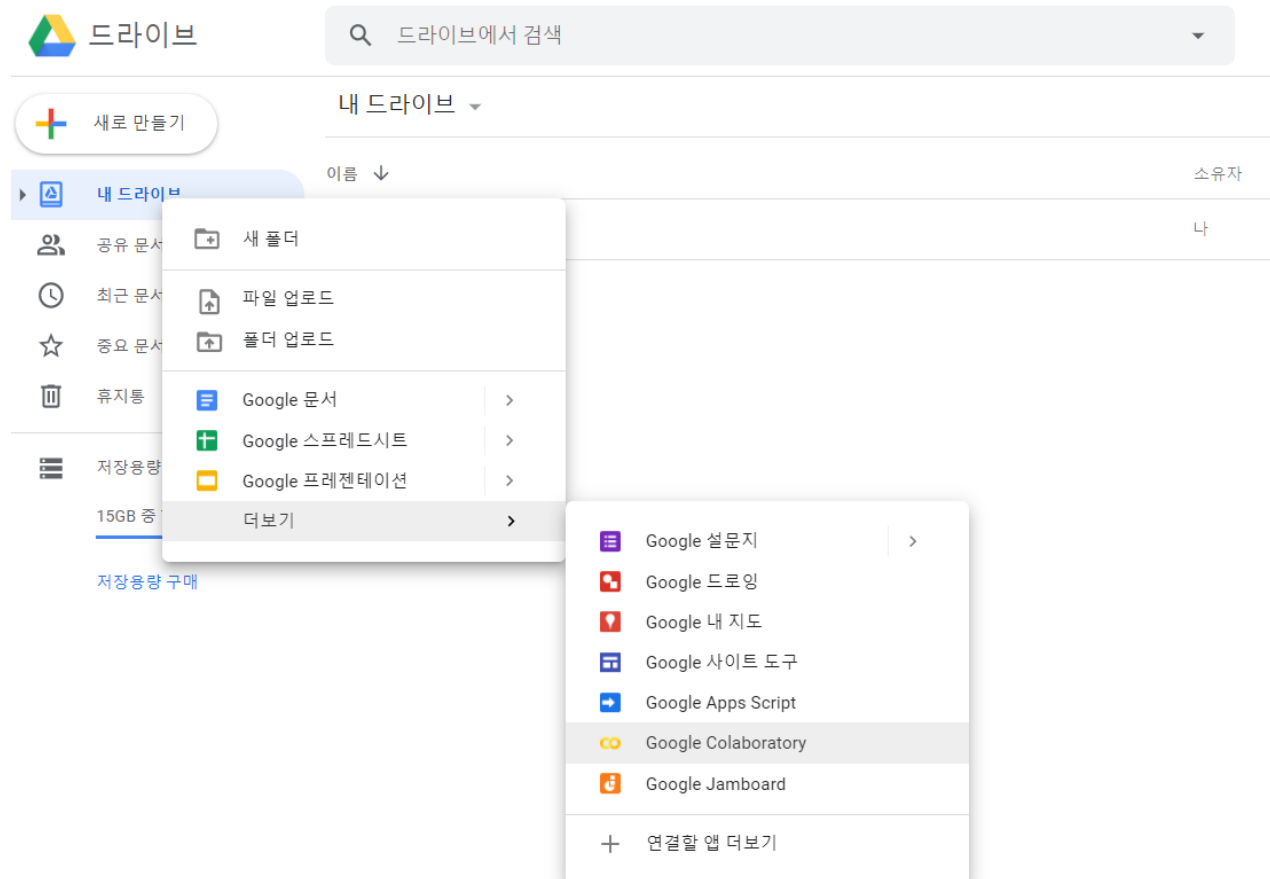
Colab 환경설정

- Colaboratory 설치 완료



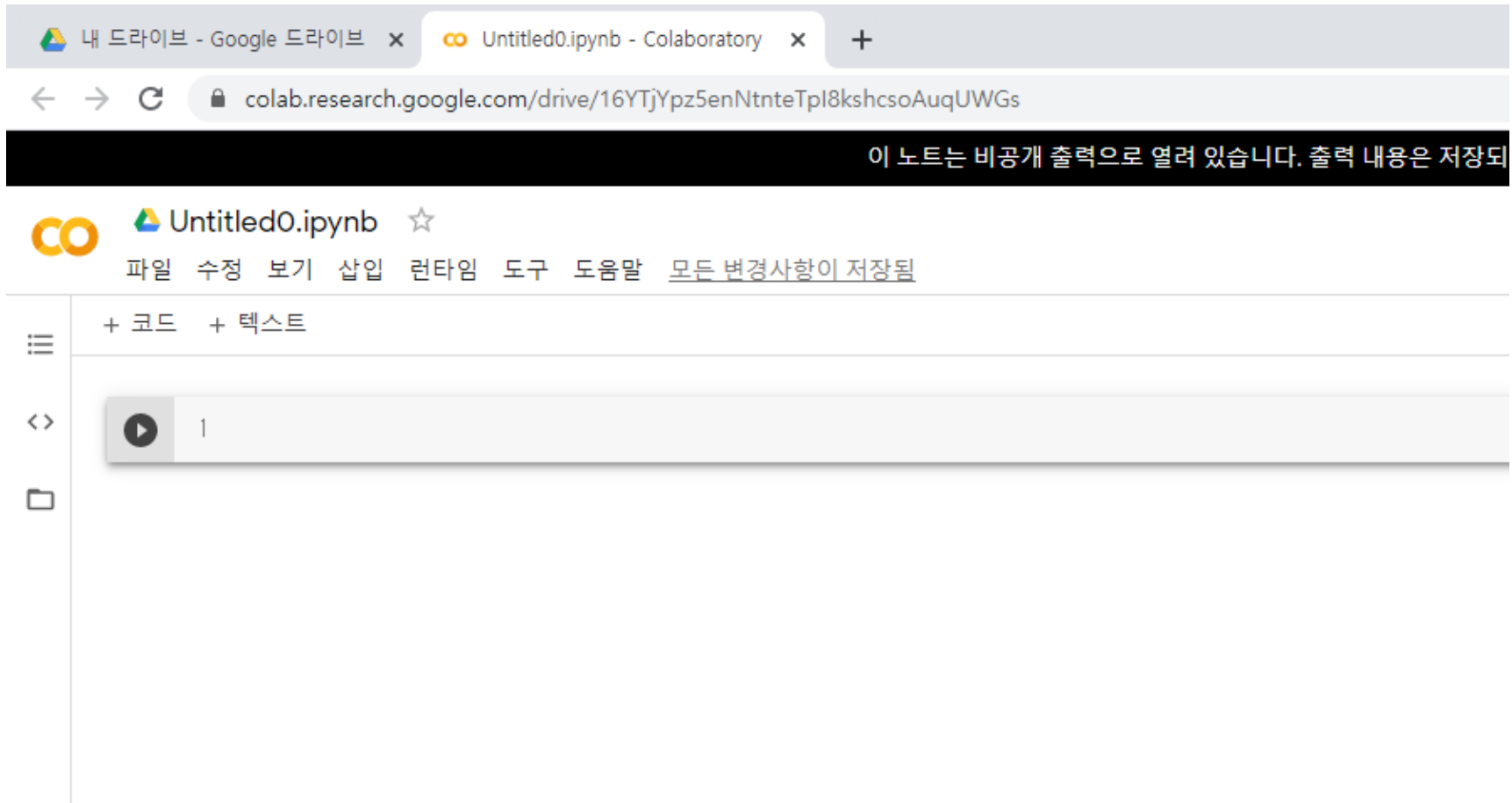
Colab 환경설정

- Colaboratory 설치 확인 및 좌클릭



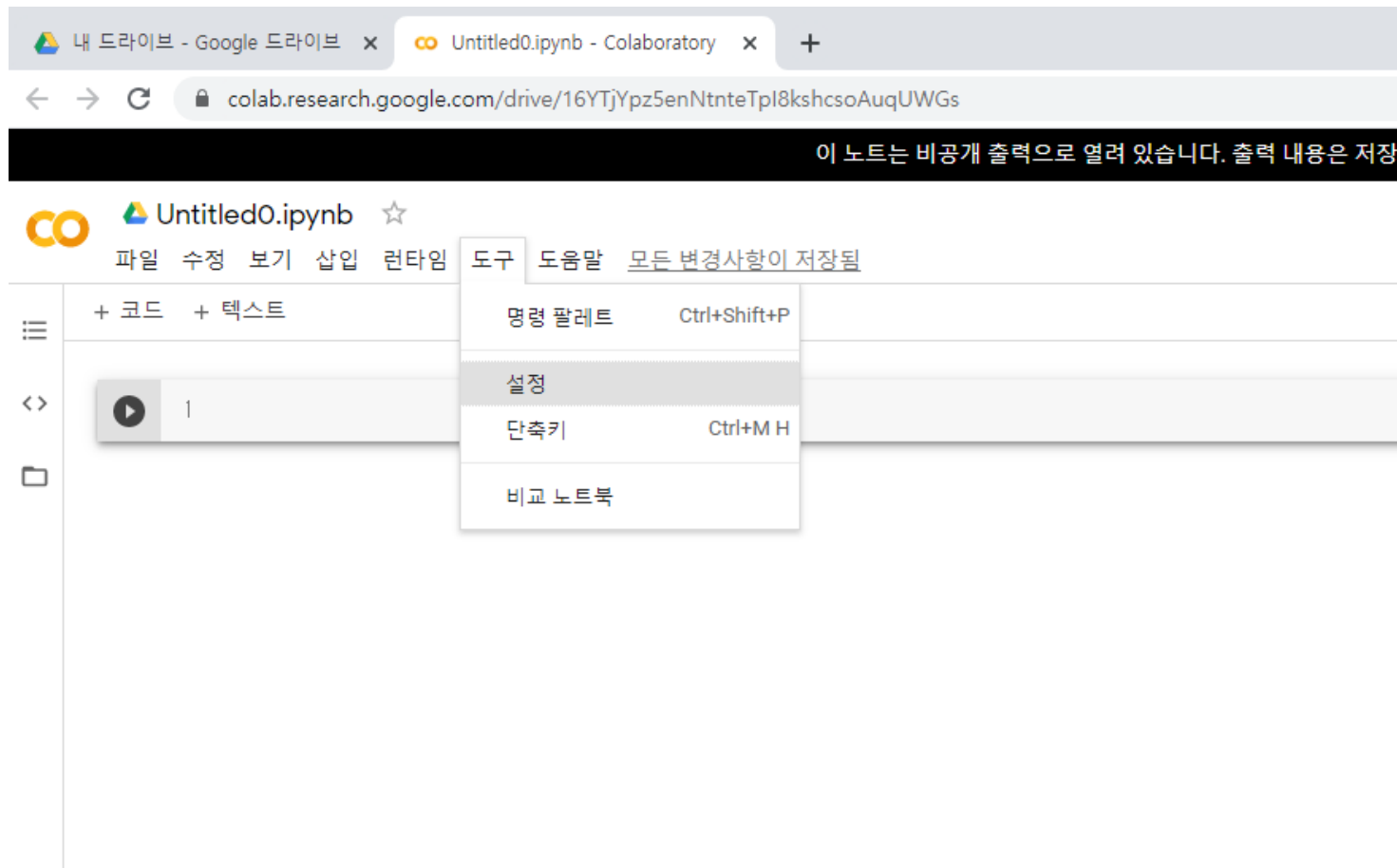
Colab 환경설정

- Chrome browser에 "Jupyter Notebook" 화면 확인



Colab 환경설정

- Chrome browser에 “Jupyter Notebook” 편집기 화면 확인



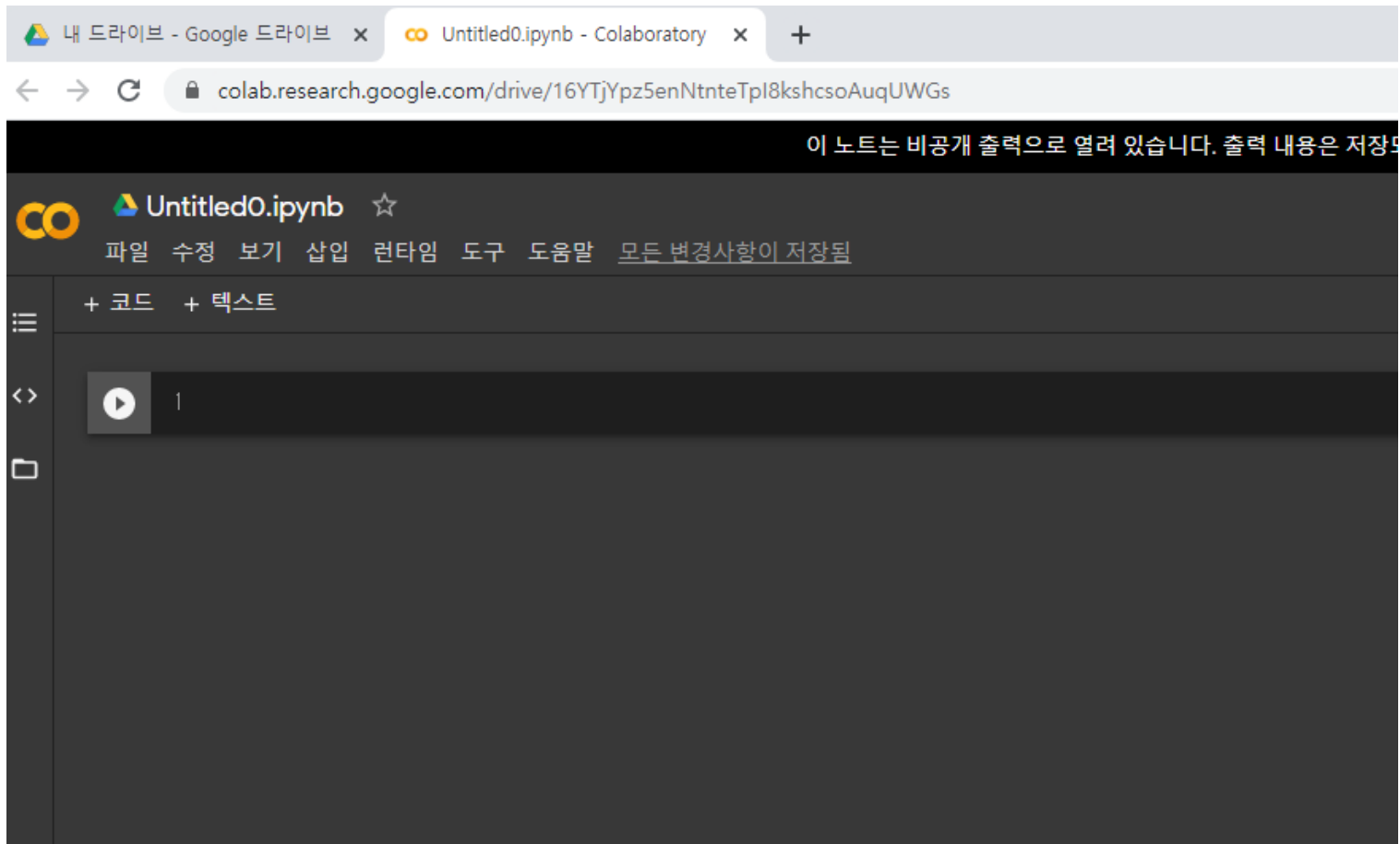
Colab 환경설정

- 편집기 화면 theme 설정(optional)
- 체크박스 클릭 확인 → 저장



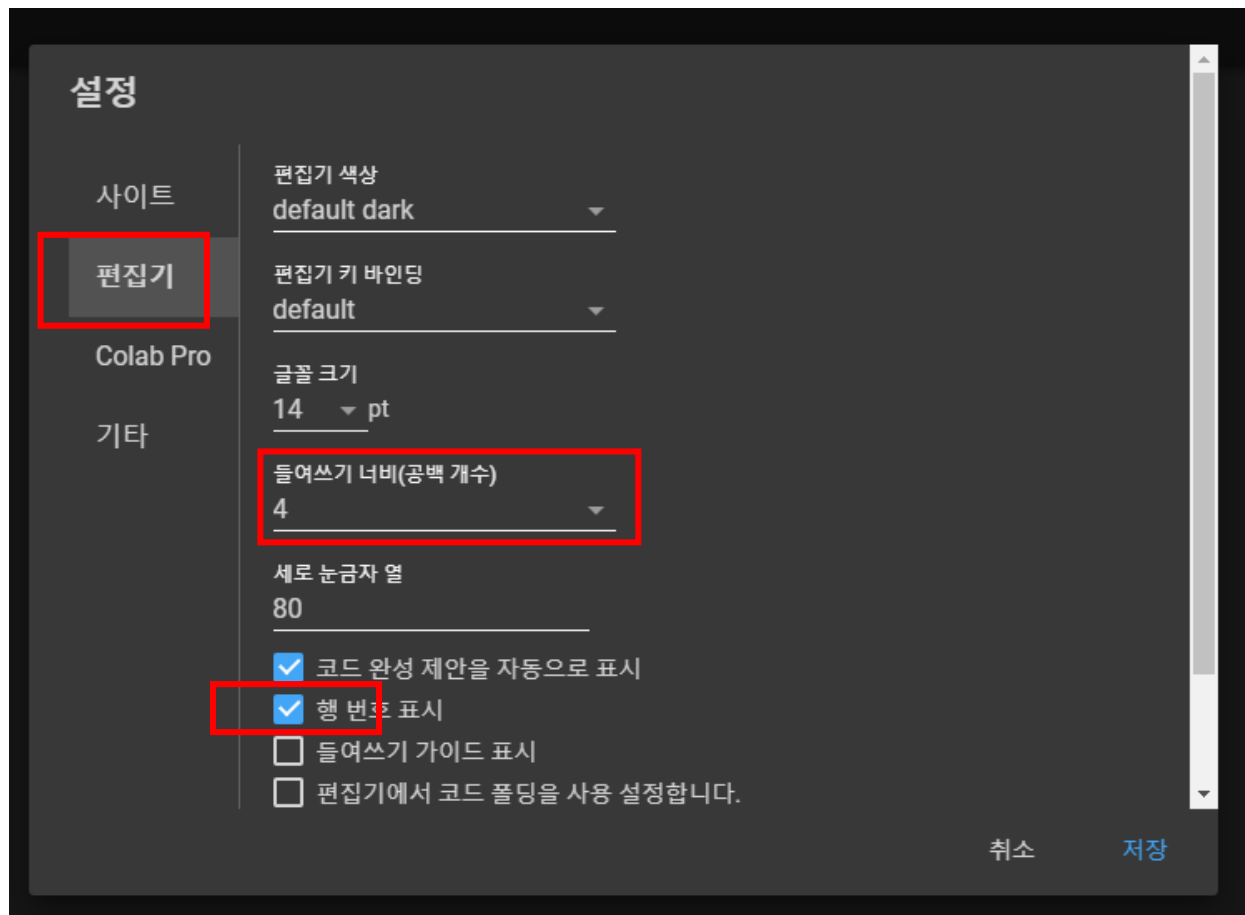
Colab 환경설정

- 편집기 화면 변경 확인



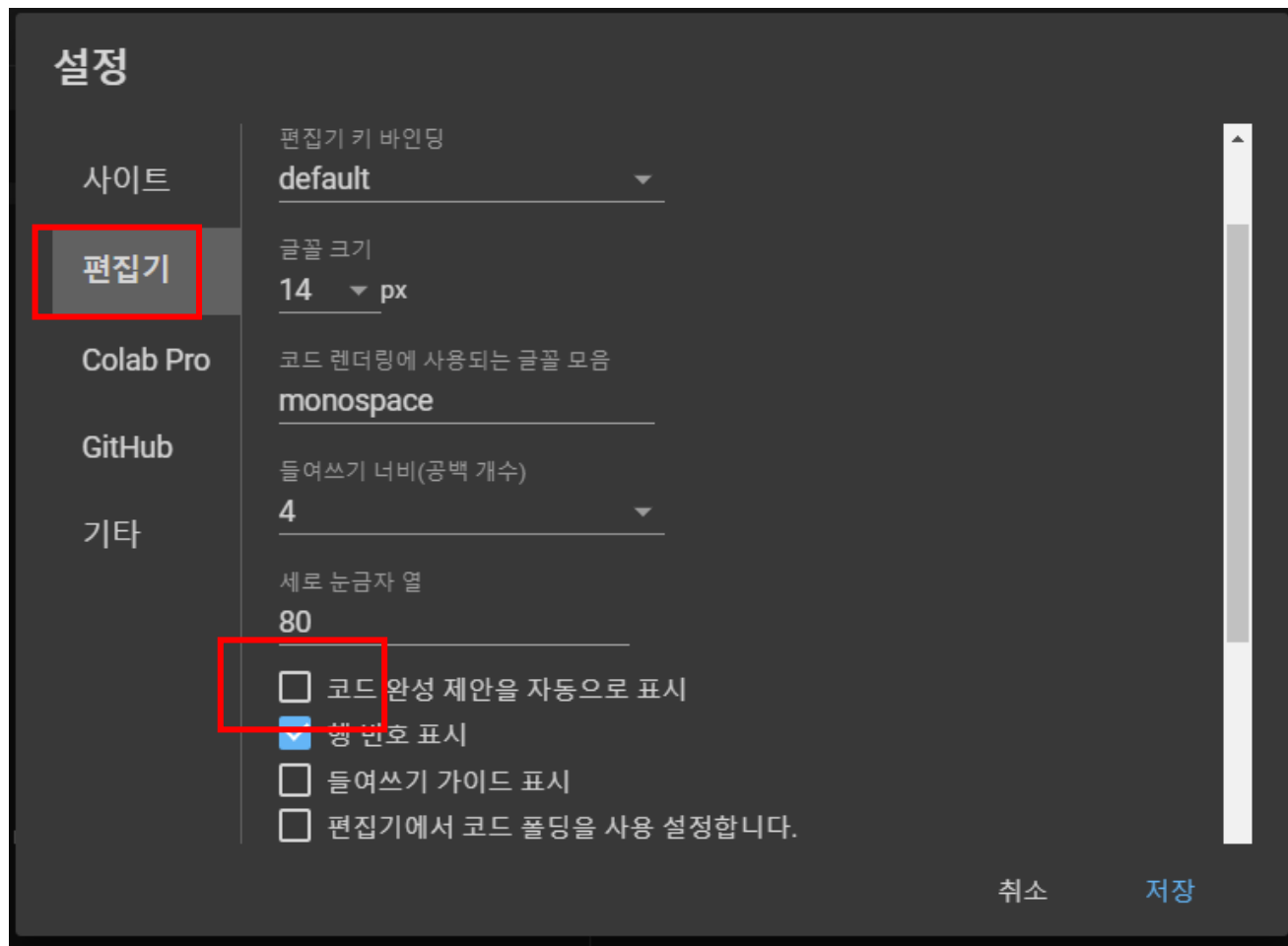
Colab 환경설정

- Python은 들여쓰기(indentation)로 구분
- Space 2칸 또는 4칸, Tab 가능하나 파이썬 코딩 스타일 가이드(PEP 8)에서는 공백 4칸으로 규정



Colab 환경설정

- 코딩 자동완성 (Docstring) 기능 활성화를 위해 "코드 완성 제안을 자동으로 표시" 체크 해제



Colab 환경설정

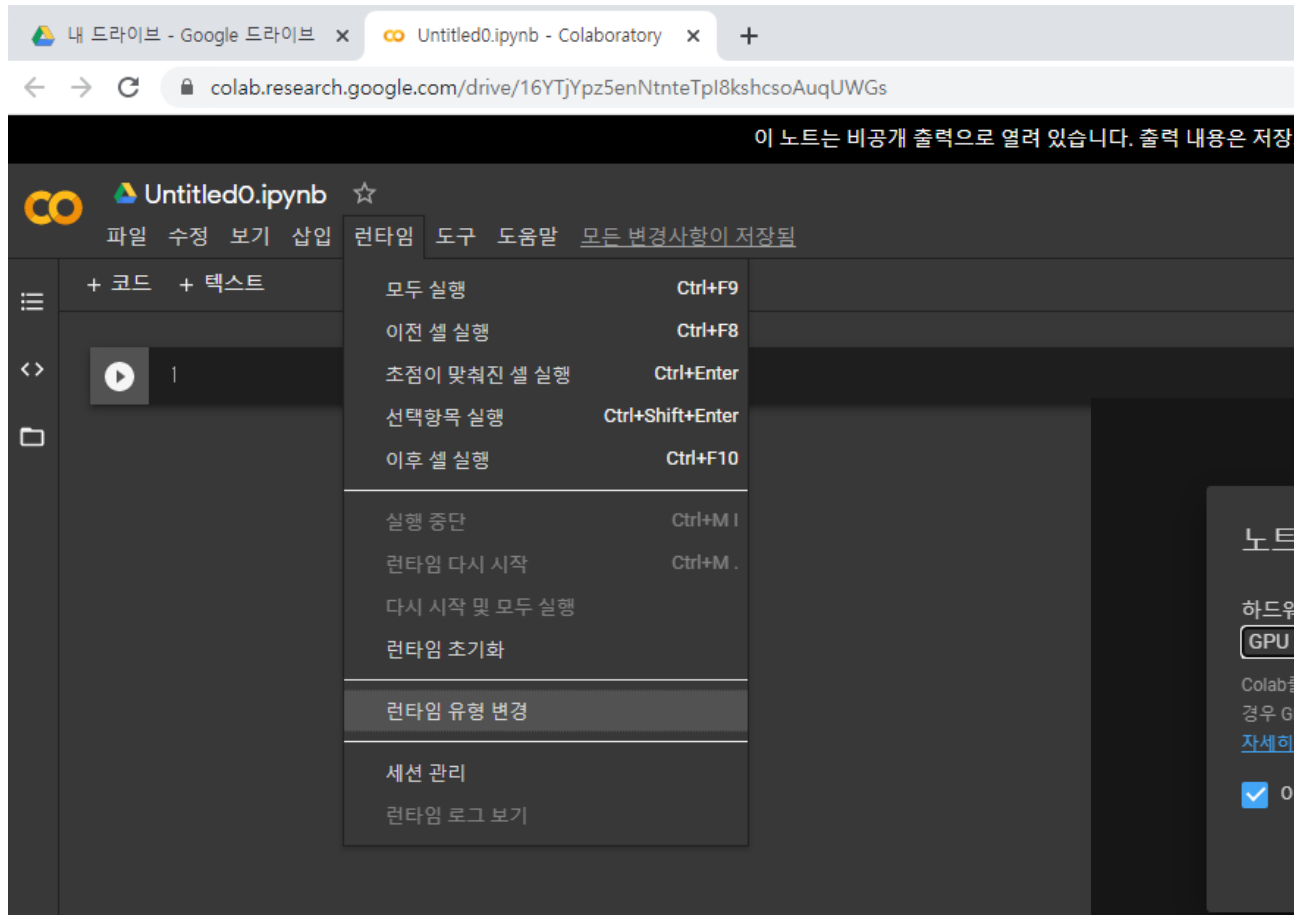
```
%config Completer.use_jedi = False
```

- Jedi 비활성화
 - % → 매직 명령어

```
1 # Import modules
2 import tenso
3     {} tensorboard
4     {} tensorboard_data_server
5     {} tensorboard_plugin_wit
6     {} tensorflow
7     {} tensorflow_datasets
8     {} tensorflow_estimator
9     {} tensorflow_gcs_config
10    {} tensorflow_hub
11    {} tensorflow_io_gcs_filesystem
12    {} tensorflow_metadata
13    {} tensorflow_probability
14    {} tensorflow as tf
15
```

Colab 환경 설정

- 런타임 유형 변경 → 하드웨어 가속기 (GPU) 선택 가능
 - 구글이 제공하는 가속기 사용 여부 저장



노트 설정

하드웨어 가속기

GPU ▼ ⓘ

Colab를 최대한 활용하려면 필요하지 않은 경우 GPU를 사용하지 않는 것이 좋습니다.

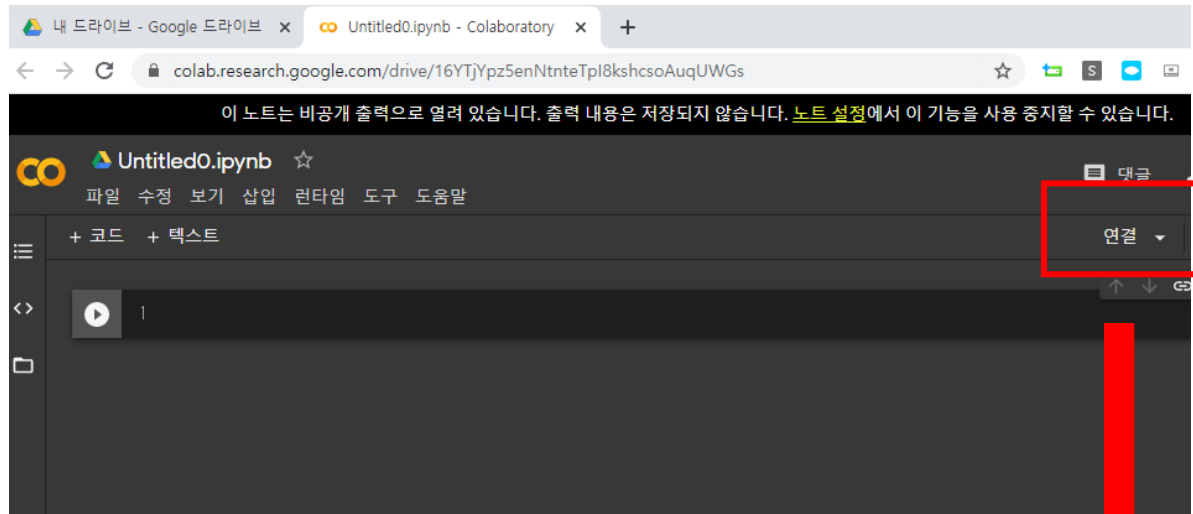
[자세히 알아보기](#)

☒ 이 노트를 저장할 때 코드 셀 출력 생략

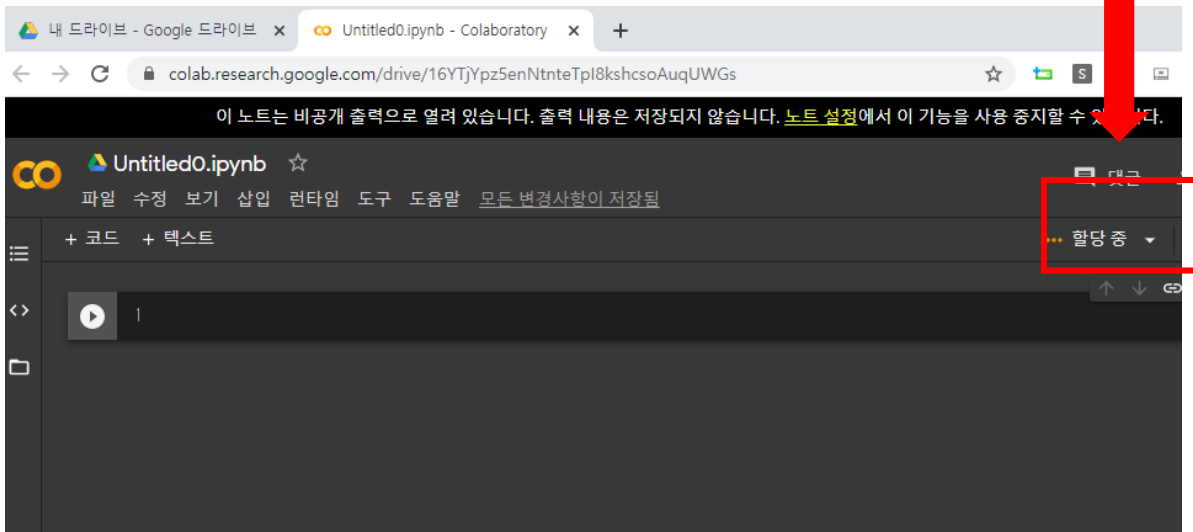
취소

저장

Colab 환경설정

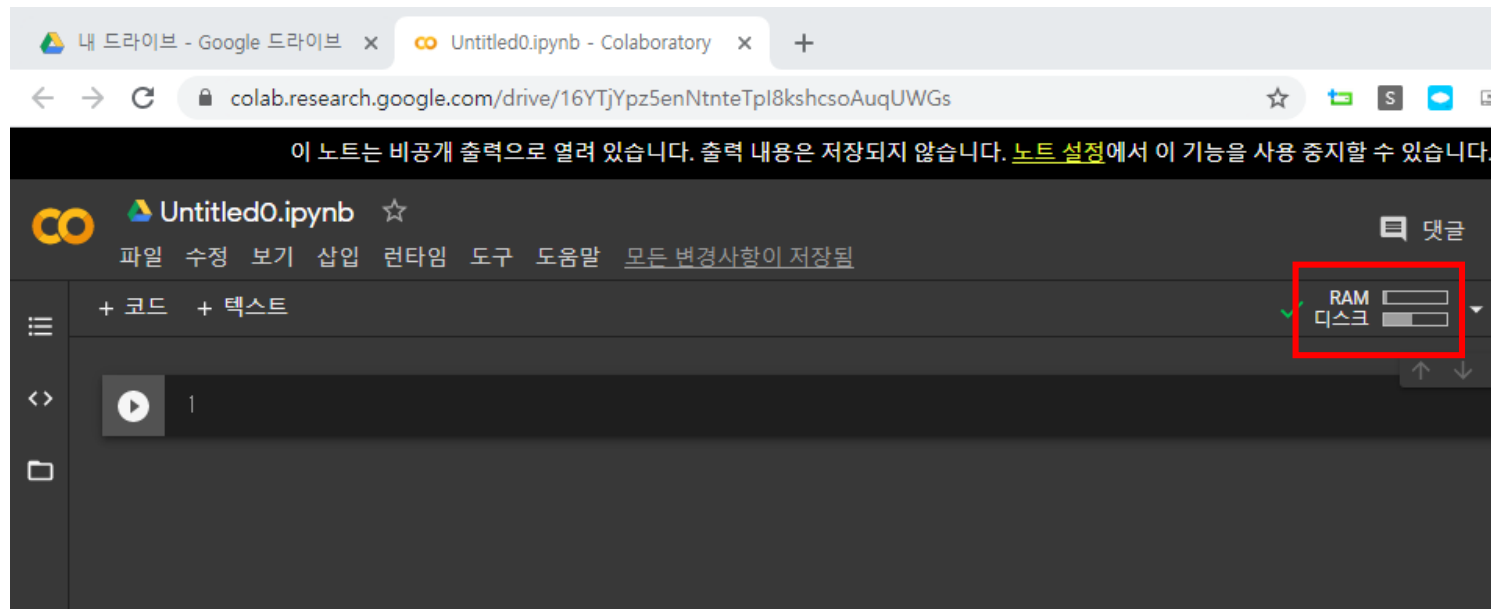


클릭



Colab 환경설정

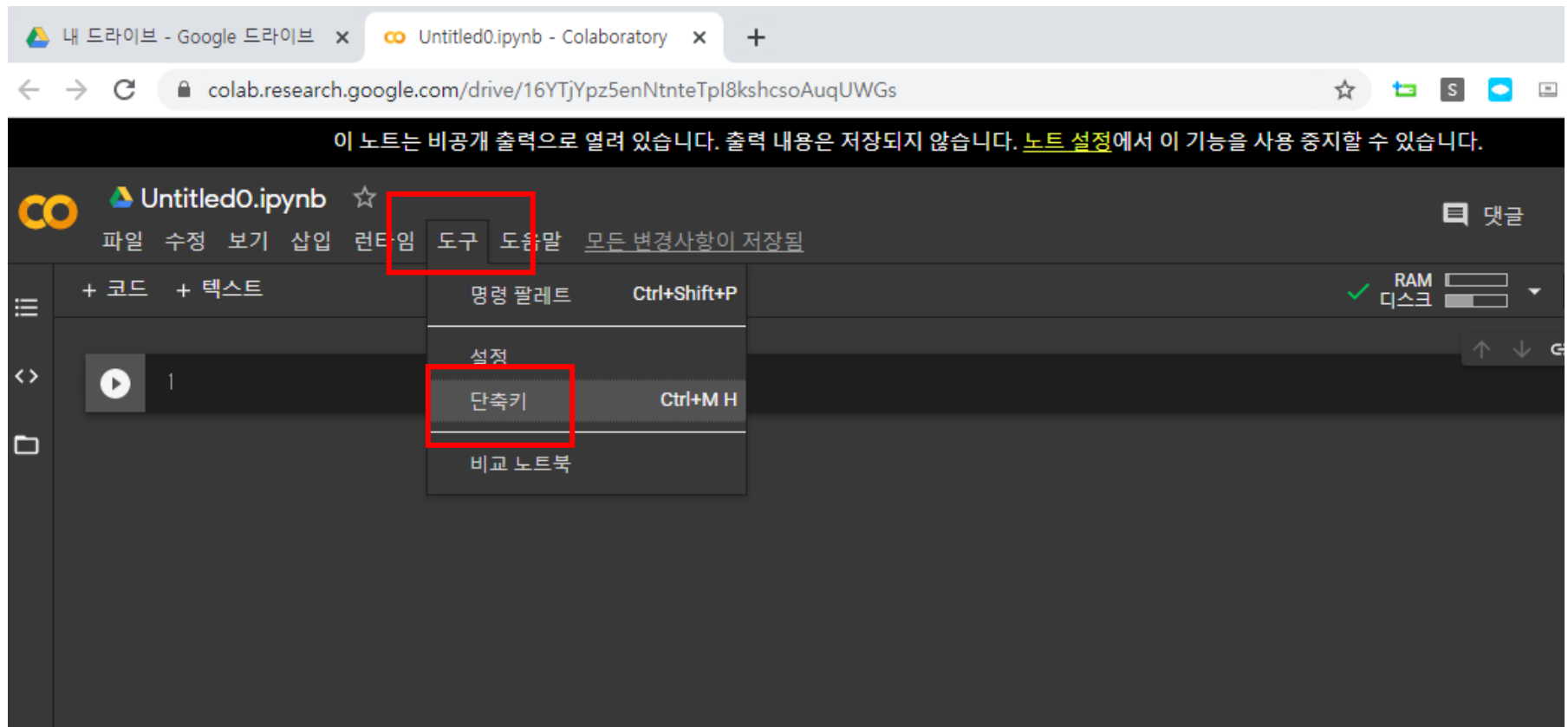
- 개발 환경 설정 완료



연결 완료!

Colab 단축키

- 도구 → 단축키 (shortcut)

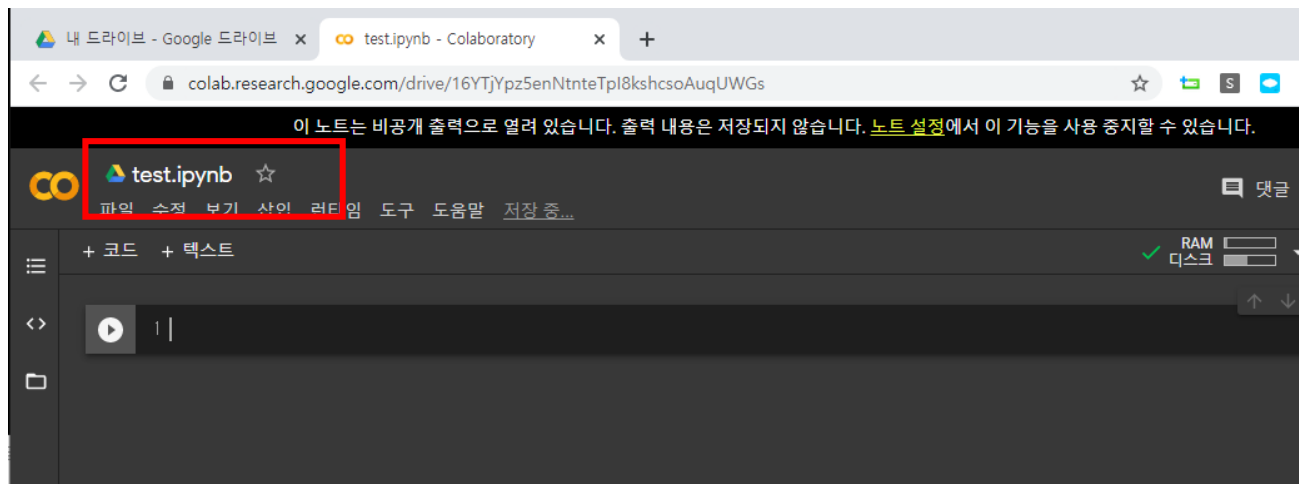
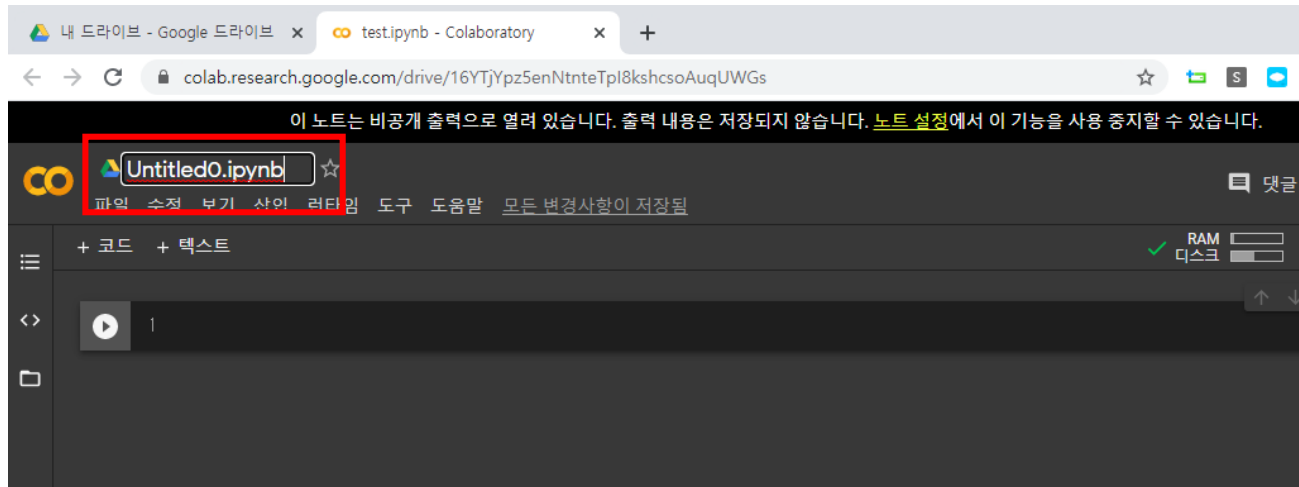


Colab 단축키

키보드 환경설정		
단축키 설정	가로형 탭 레이아웃 보기	Ctrl+Shift+Y 셀 작업 다시 실행
단축키 설정	노트 공유	Ctrl+M D 셀/선택항목 삭제
단축키 설정	노트 설정 열기	Shift+Enter 셀을 실행하고 다음 셀 선택
단축키 설정	노트 업로드	Alt+Enter 셀을 실행하고 새 셀 삽입
Ctrl+O	노트 열기	단축키 설정 스크래치 셀로 복사
Ctrl+P	노트 인쇄	Ctrl+Alt+N 스크래치 코드 셀 열기
Ctrl+S	노트 저장	단축키 설정 실행 모드에서 열기
단축키 설정	노트 출처 보기	Ctrl+M I 실행 중단
단축키 설정	노트 파일 정보 표시	단축키 설정 실행된 코드 기록 표시
단축키 설정	노트를 휴지통으로 이동	Ctrl+M B 아래에 코드 셀 삽입
Ctrl+F9	노트의 모든 셀 실행	Ctrl+M F 양식 보기 순환
Ctrl+M N	다음 셀	단축키 설정 양식 입력란 추가
Shift+Down	다음 셀을 포함하도록 선택 영역을 확장합니다.	단축키 설정 양식 추가
Ctrl+Shift+]	다음 탭으로 포커스 이동	Ctrl+M A 위에 코드 셀 삽입
Shift+Click	다중 셀 선택	단축키 설정 의견 사이드바
단축키 설정	단일 탭 레이아웃 보기	Ctrl+M P 이전 셀
Ctrl+M H	단축키 표시	Shift+Up 이전 셀을 포함하도록 선택 영역을 확장합니다.
단축키 설정	댓글 창 열기	Ctrl+Shift+[이전 탭으로 포커스 이동
Ctrl+Alt+M	댓글 추가	Ctrl+Space 또는 Tab 자동완성 ?
단축키 설정	드라이브 마운트	Ctrl+G 전체 다음 찾기
단축키 설정	드라이브 마운트 해제	Ctrl+Shift+G 전체 이전 찾기
단축키 설정	드라이브에 사본 저장	Ctrl+H 전체 찾기/바꾸기
단축키 설정	드라이브에서 찾기	단축키 설정 접힌 섹션 레이아웃 저장
Ctrl+M .	런타임 다시 시작	

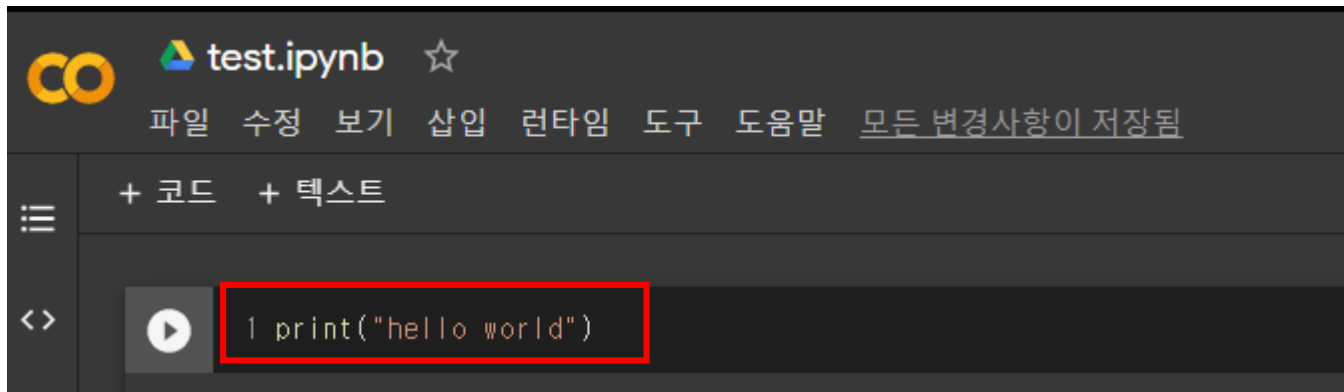
Colab 실행

- ipynb 파일 이름 변경

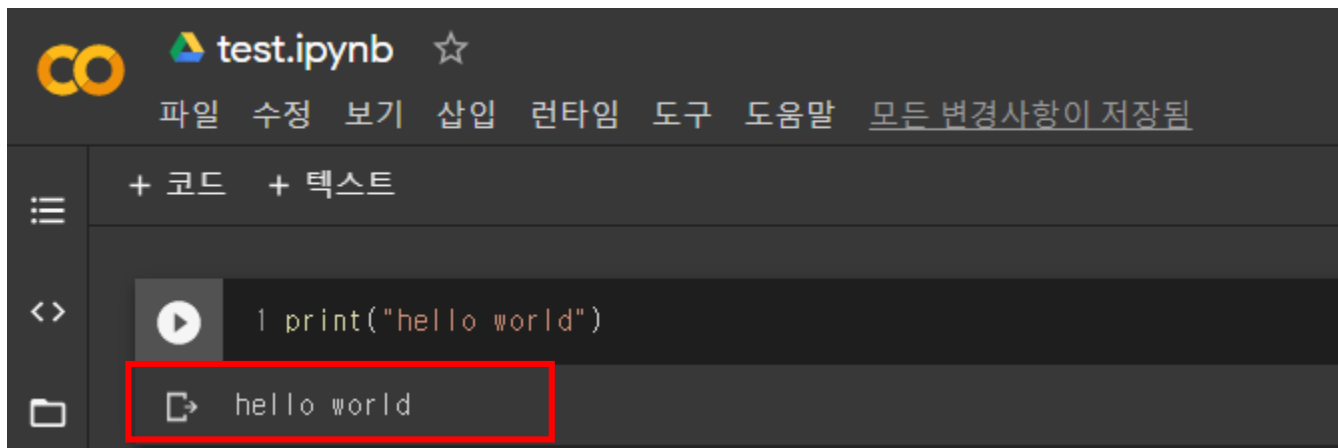


Python script 작성 및 실행

- `print("hello world")` 라는 script 작성

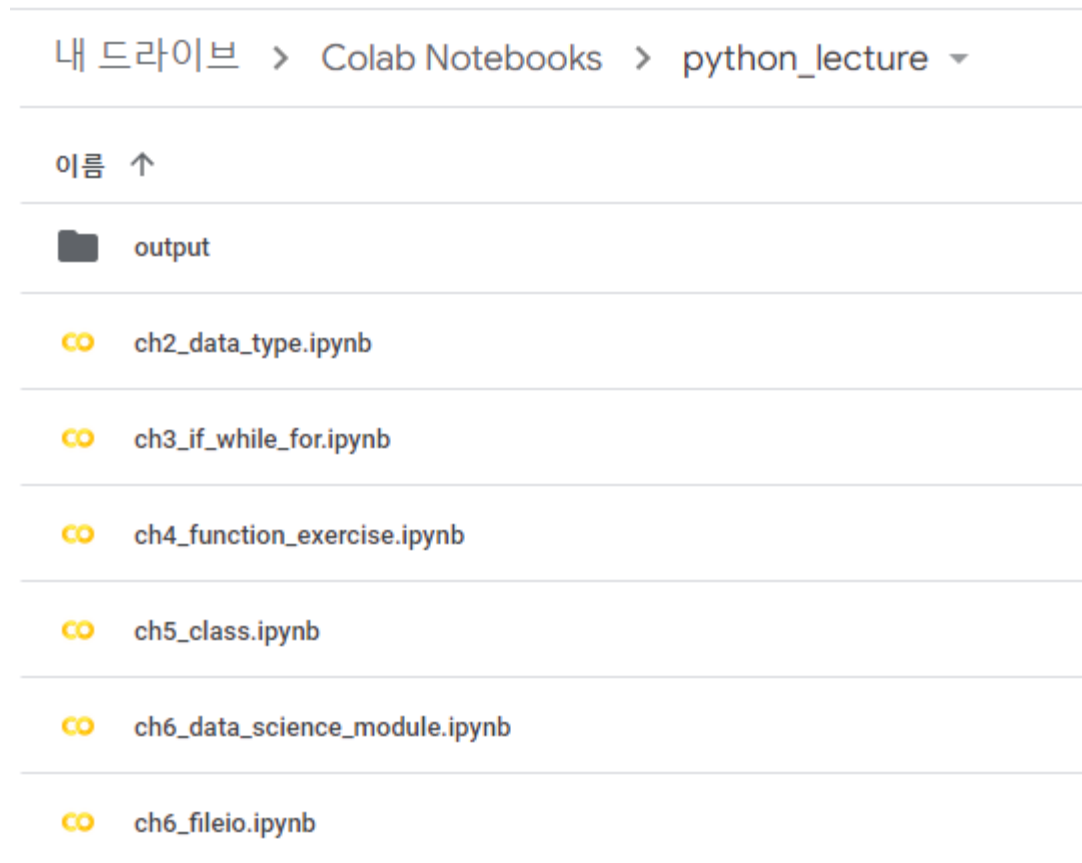


- Ctrl + Enter 단축키로 cell 실행 (run)



Ipynb 파일 저장

- Google drive에 "파일명.ipynb"로 저장 확인





Next Class

- 프로그래밍 기초 (메모리, 변수)
- Python 기초 문법