

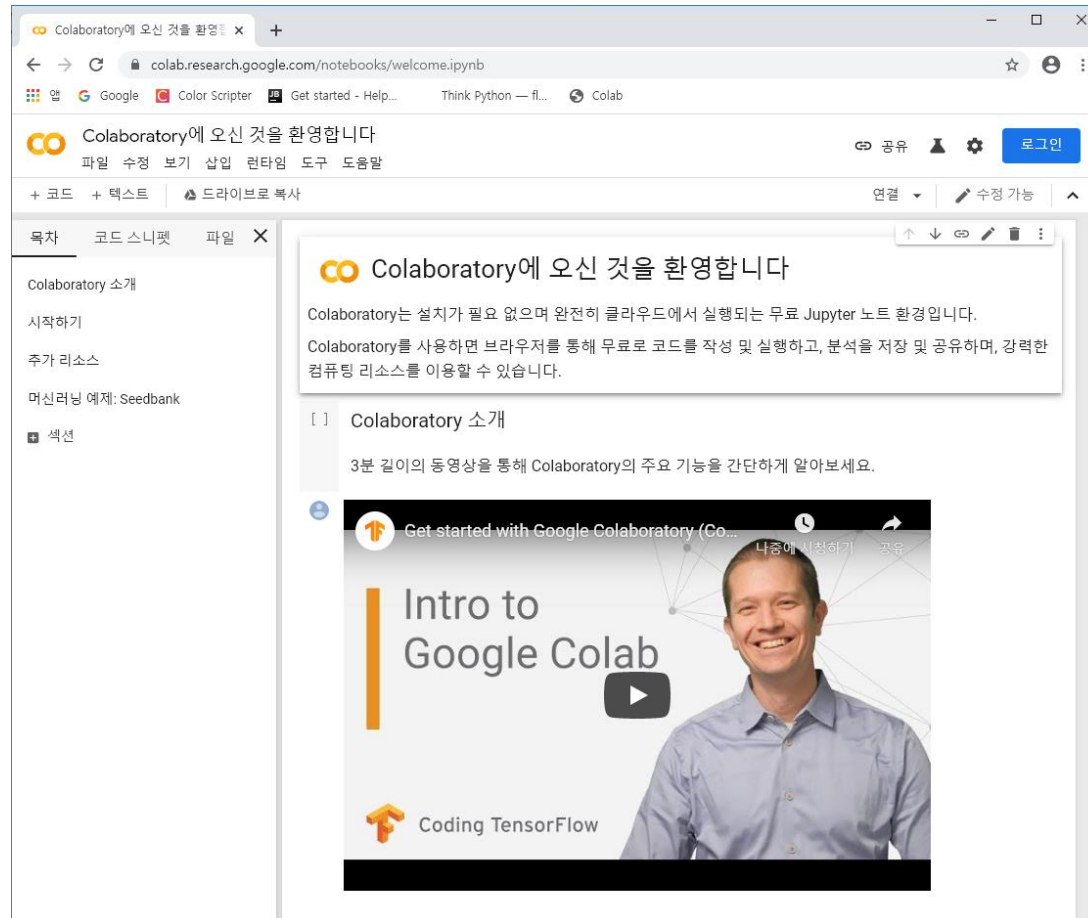
구글의 Colab 개요와 둘러보기

구글 코랩 Google Colaboratory

- **클라우드 기반의 무료 Jupyter 노트북 개발 환경**
 - 주피터 노트북을 지원하는 머신러닝, 딥러닝 클라우드 개발환경
 - 파이썬 뿐만 아니라 판다스, 멧플롯리브의 시각화 및 텐서플로우나 케라스 등 딥러닝 라이브러리도 쉽게 사용
 - <https://colab.research.google.com>
- **Google Drive + Jupyter Notebook**
 - 구글 계정 전용의 가상 머신 지원 – GPU, TPU 지원
 - Google drive 문서와 같이 링크만으로 접근 / 협업 가능
 - 구글 계정 필요
- **장점**
 - 구글 드라이브와 연계
 - 기본적으로 폴더 Colab Notebooks과 연결
 - 깃허브와 연계
 - 깃허브 소스를 바로 코딩 가능
 - *.ipynb

코랩 접속

- <https://colab.research.google.com/notebooks/welcome.ipynb>



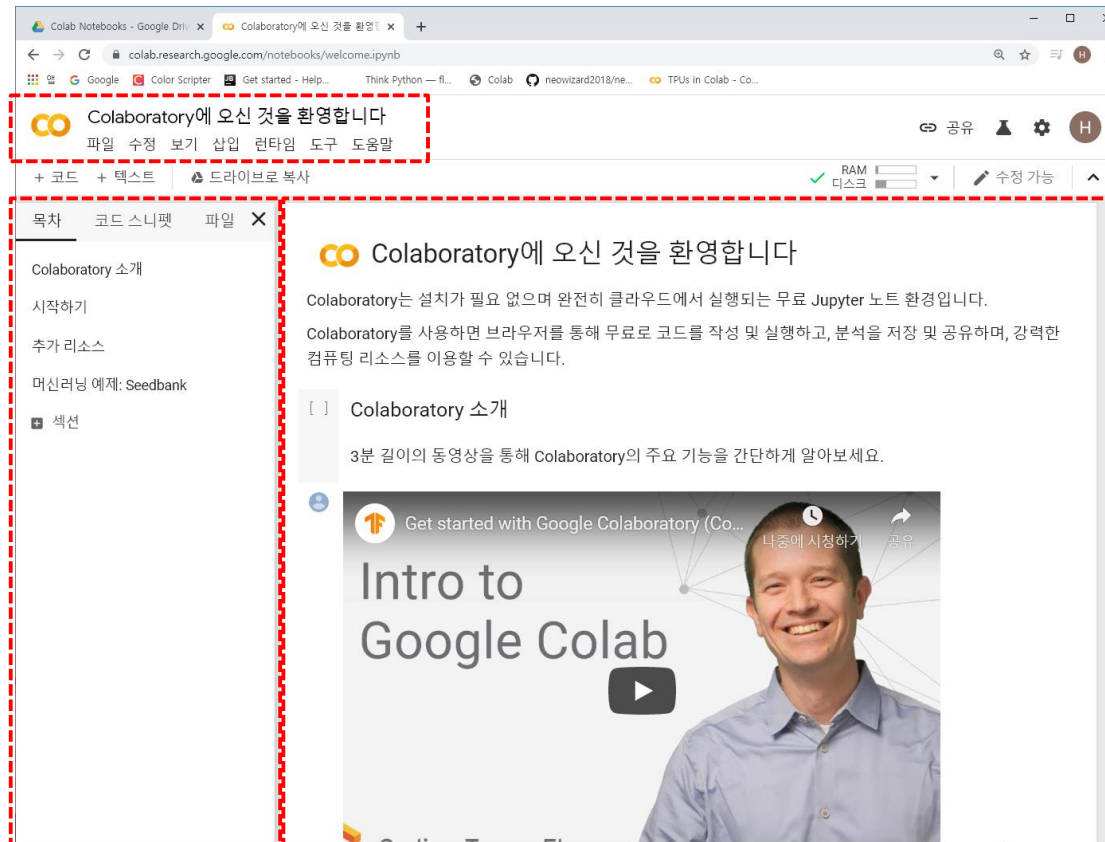
코랩 서버의 사양

- 일반 개인 PC보다 성능이 우수
 - CPU: Intel Xeon 2.2 GHz
 - RAM: 13GB
 - 저장공간: 33GB
 - 90분간 미사용 시 중지
 - 최대 12시간 연속 사용 가능
- CPU 사용
 - 생각보다 많이 빠르지는 않음
- 딥러닝은 GPU, TPU를 사용, 상당히 좋음
 - Graphics Processing Unit
 - Tensor Processing Unit

구글 Colab Welcome

Welcome 노트

- 브라우저에서 Colab 주소를 입력
 - 대화상자 외부 클릭
 - 메뉴, 목차, 본문인 노트로 구성



영상과 코드 셀 실행

• 목차

- Colaboratory 소개, 시작하기, 추가 리소스, 머신러닝 예제

• 환영인사

- Colaboratory는 설치가 필요 없으며 완전히 클라우드에서 실행되는 무료 Jupyter 노트 환경입니다.
- Colaboratory를 사용하면 브라우저를 통해 무료로 코드를 작성 및 실행하고, 분석을 저장 및 공유하며, 강력한 컴퓨팅 리소스를 이용할 수 있습니다.

• 자막을 키고 영상 시청

• 코드 셀 실행

- 버튼
- Ctrl+Enter
- Shift+ Enter

- 실행 후
다음 셀로
이동



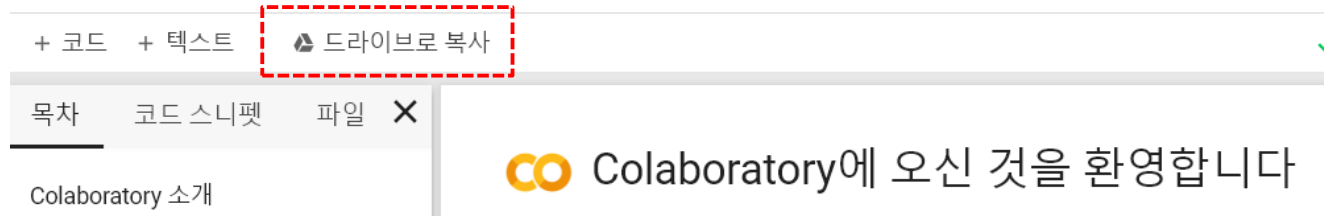
위 셀의 코드를 실행하려면 셀을 클릭하여 선택한 후, 코드 왼쪽의 실행 버튼을 누르거나 단축키 'Command/Ctrl+Enter'를 사용합니다.

모든 셀은 동일한 전역 상태를 수정하므로 셀을 실행하여 정의되는 변수는 다른 셀에서도 사용할 수 있습니다.

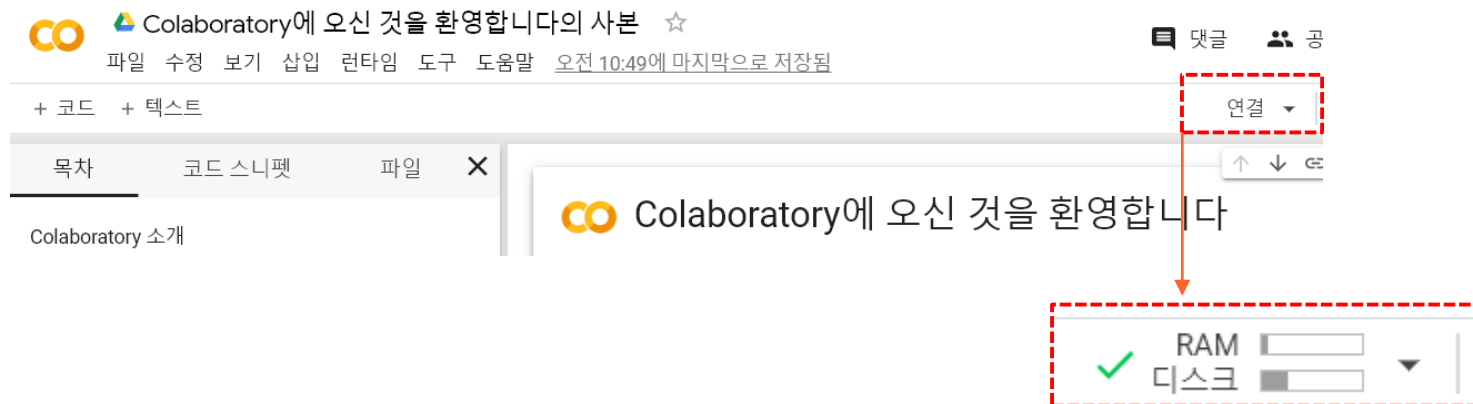
파일 저장도 가능

• 버튼

- 드라이버로 복사

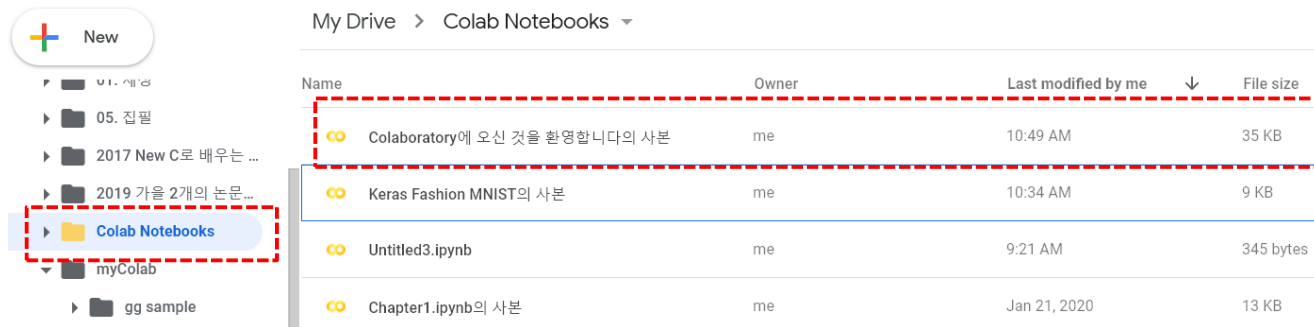


- 새로운 사본이 생성되며, 버튼 '드라이브로 복사'가 사라짐
 - **연결로 다시 시작**

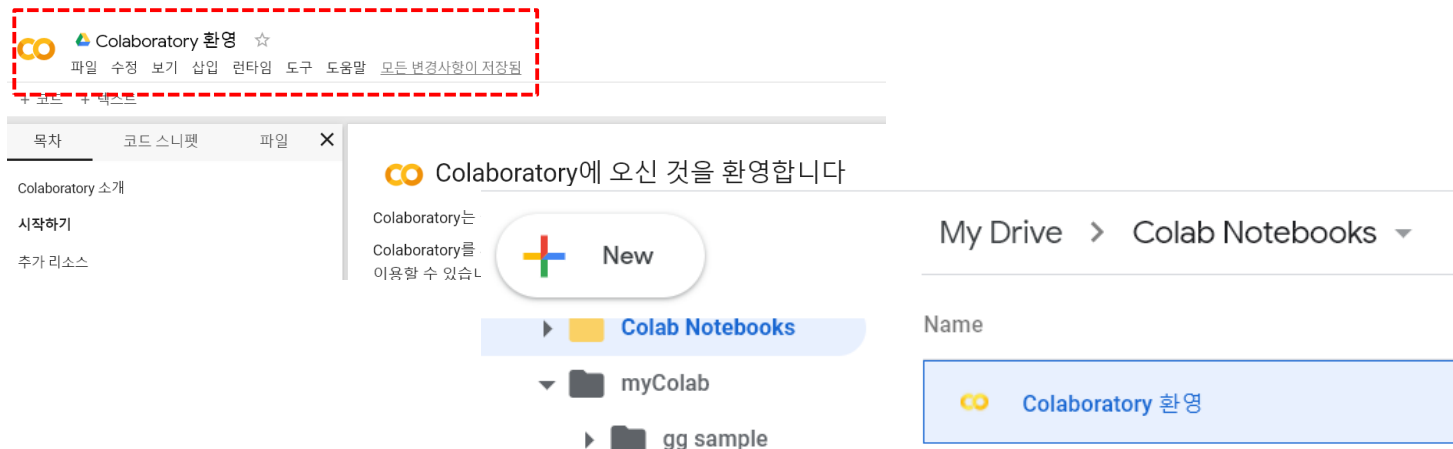


자신의 구글 드라이브 확인

- 폴더 Colab Notebooks가 생성
 - 하부에 파일 확인 가능



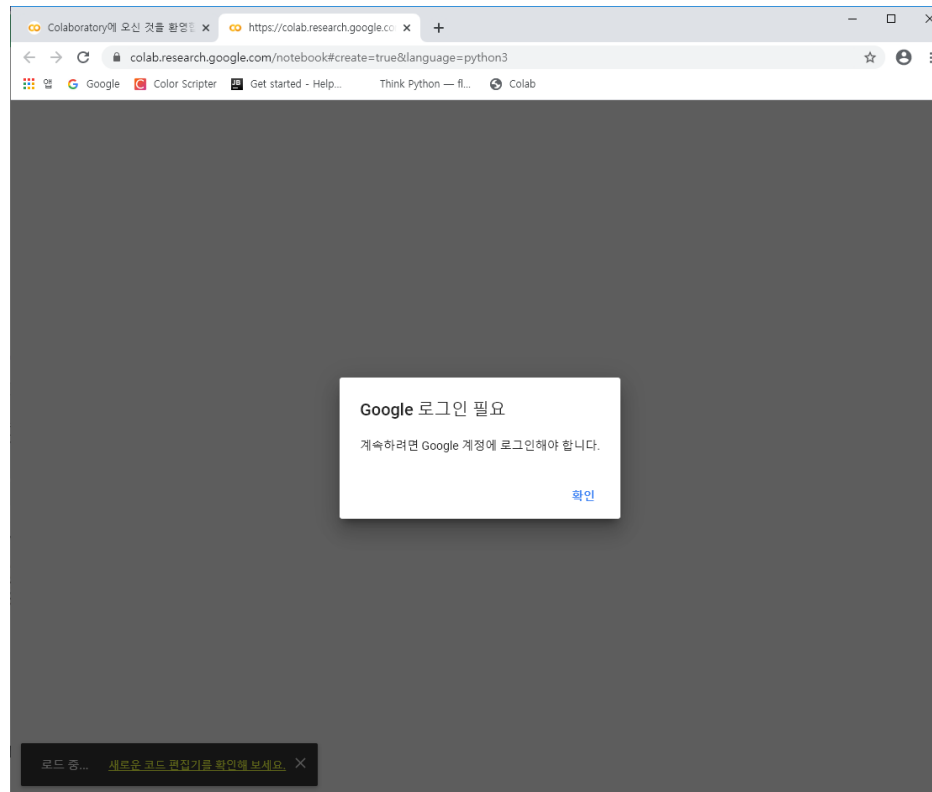
- 이름 수정 후 저장



직접 노트 파일 만들기

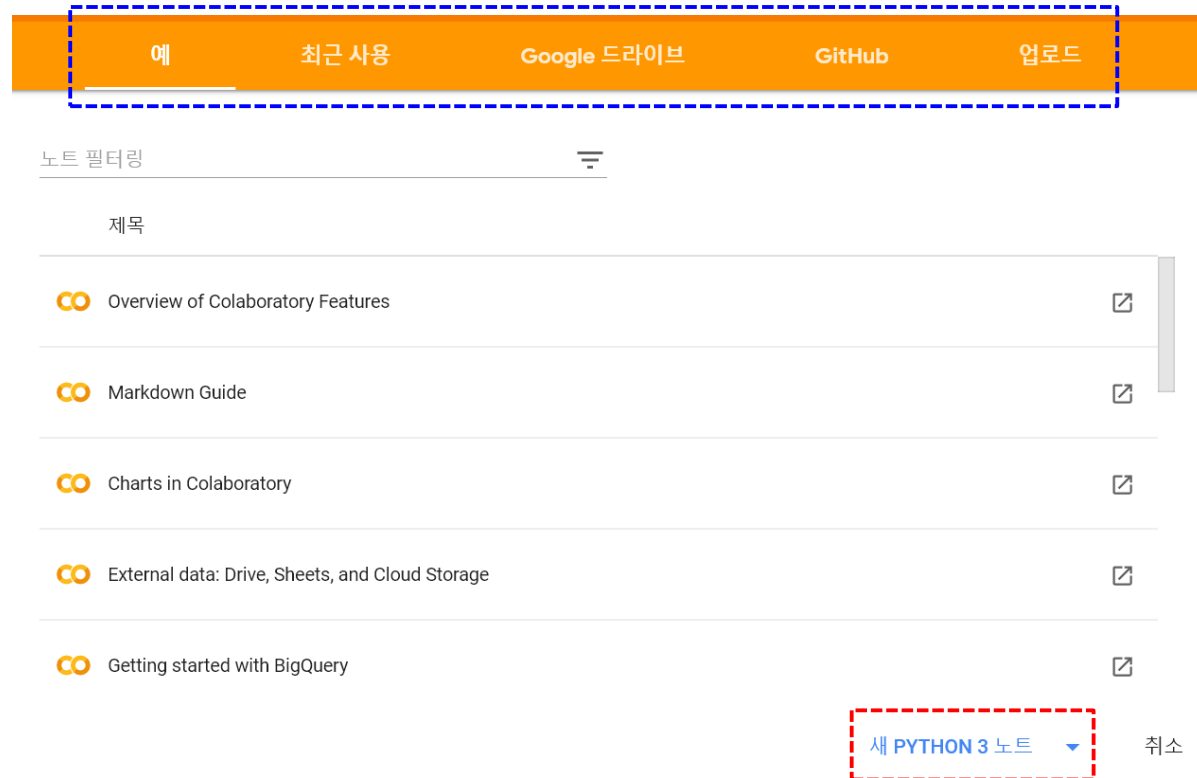
직접 노트 파일 생성과 작성

- 메뉴 파일, 새 Python3 노트
 - 로그인 필요



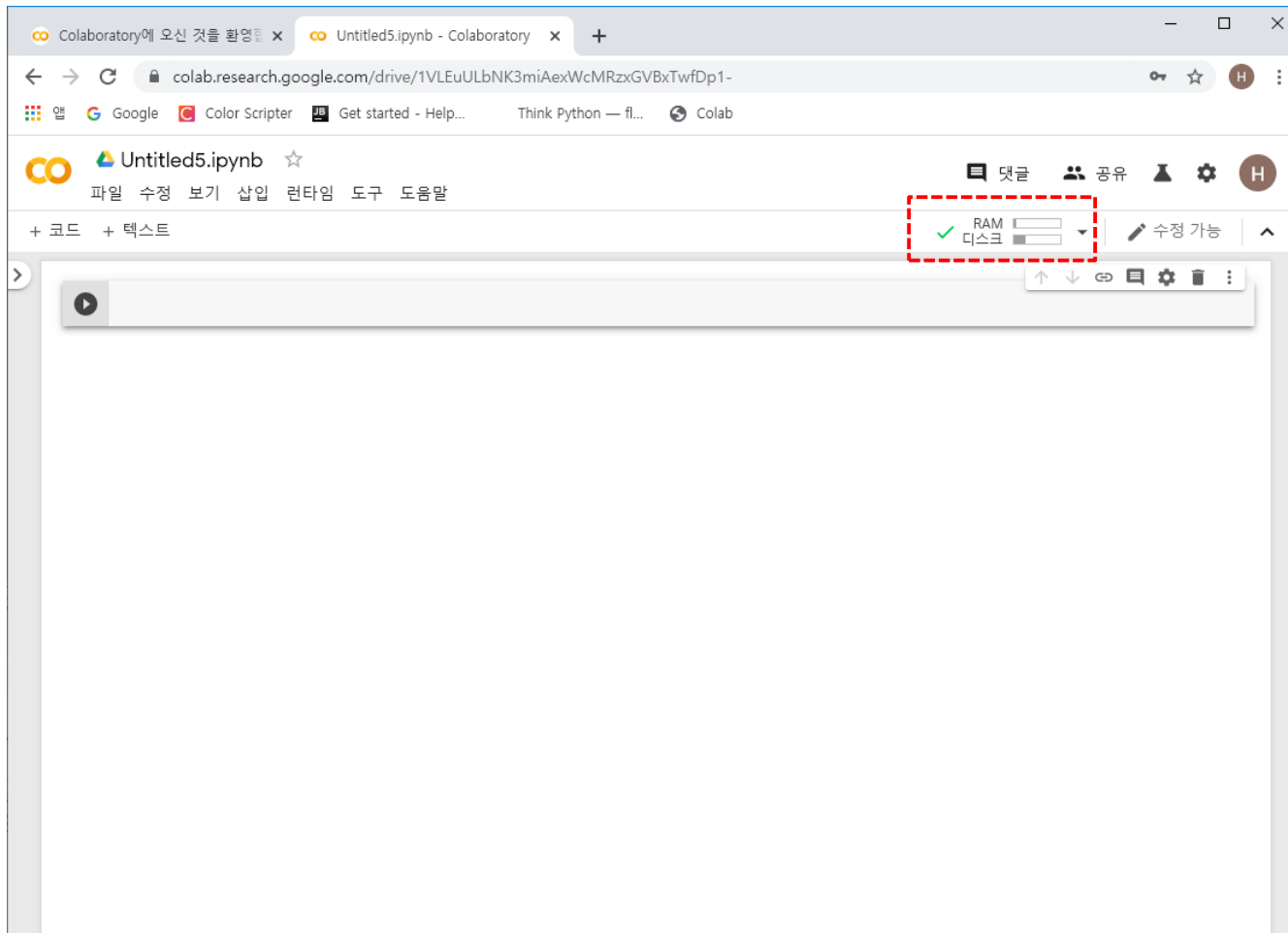
다양한 방법의 노트 접근

- 예, 최근 사용, Google 드라이브, GitHub, 업로드
- 무조건 노트 파일 작성
 - 새 PYTHON3 노트 선택



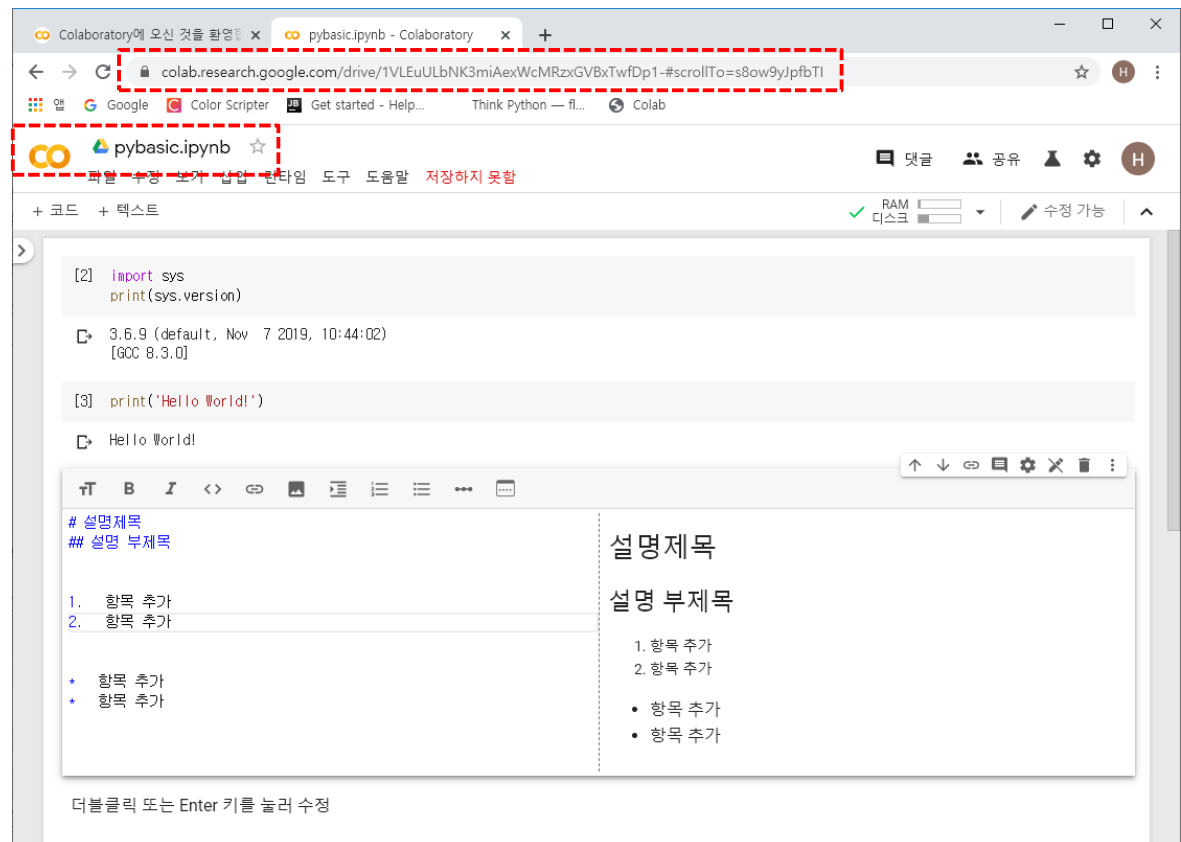
제일 먼저 서버 연결

- 우측 상단 '연결' 클릭



노트 파일 편집

- 셀
 - 문서와 코드의 단위
 - 코드 셀과 텍스트 셀
 - 실행
 - Shift + enter
- 파일 이름 수정
 - 좌측 상단
 - 확장자: *.ipynb
 - Interactive
 - PYthon
 - Note Book
- url 활용
 - 외부 공유 가능



Colab 편집

• 파일 이름 변경



• Code cell, Text cell

- .ipynb 파일은 code cell과 text cell로 구성
- 각 셀 하단에 마우스를 대거나, 화면 좌상단 버튼으로 셀 추가 가능
- 셀 선택(마우스) 후 셀 우상단 삭제 버튼으로 셀 삭제 가능

Colab Code Cell

• 코드 박스

- 일반적인 파이썬 코딩 방식과 동일
- 각 셀은 한번에 실행할 단위를 뜻함
- 실행 이후에도 메모리는 유지되어 다른 셀 실행 시 영향을 줌
 - 런타임 다시 시작 시 초기화

• 상단 메뉴의 런타임

- 실행 중인 셀 중단
- 런타임 다시 시작

The image contains two screenshots of the Google Colab interface with various elements highlighted by yellow boxes and labels.

Top Screenshot:

- 실행 번호 (Execution Number):** A box labeled "실행 번호" points to the cell number "[1]" in the top-left corner of the code cell.
- 실행 (Ctrl + Enter) (Run):** A box labeled "실행 (Ctrl + Enter)" points to the play button icon on the left side of the code cell.
- 실행 결과 (Execution Result):** A box labeled "실행 결과" points to the output area below the code cell, which displays the number "3".

Bottom Screenshot:

- 런타임 (Runtime):** A box labeled "런타임" points to the "런타임" (Runtime) menu item in the top navigation bar.
- 실행 중단 런타임 재시작 (Stop Runtime and Restart):** A box labeled "실행 중단 런타임 재시작" points to the "실행 중단" (Stop Runtime) and "런타임 다시 시작..." (Restart Runtime...) options in the Runtime menu.

The code in the top cell is:

```
[1] # Code Cell!
a = 1
b = 2
print(a+b)

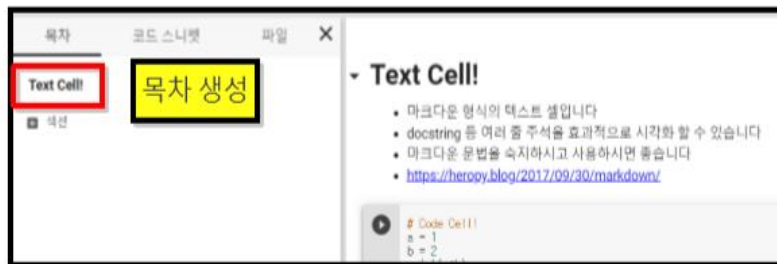
# Ctrl+Enter 로 해당 코드 셀 실행
```

The code in the bottom cell is:

```
# 각 셀은 한번에 실행할 단위를 뜻함
# 실행 이후에도 메모리는 그대로 유지되어 다른 셀의 실행에 영향을 줌
a += 3
b -= 1
print(a+b)
```

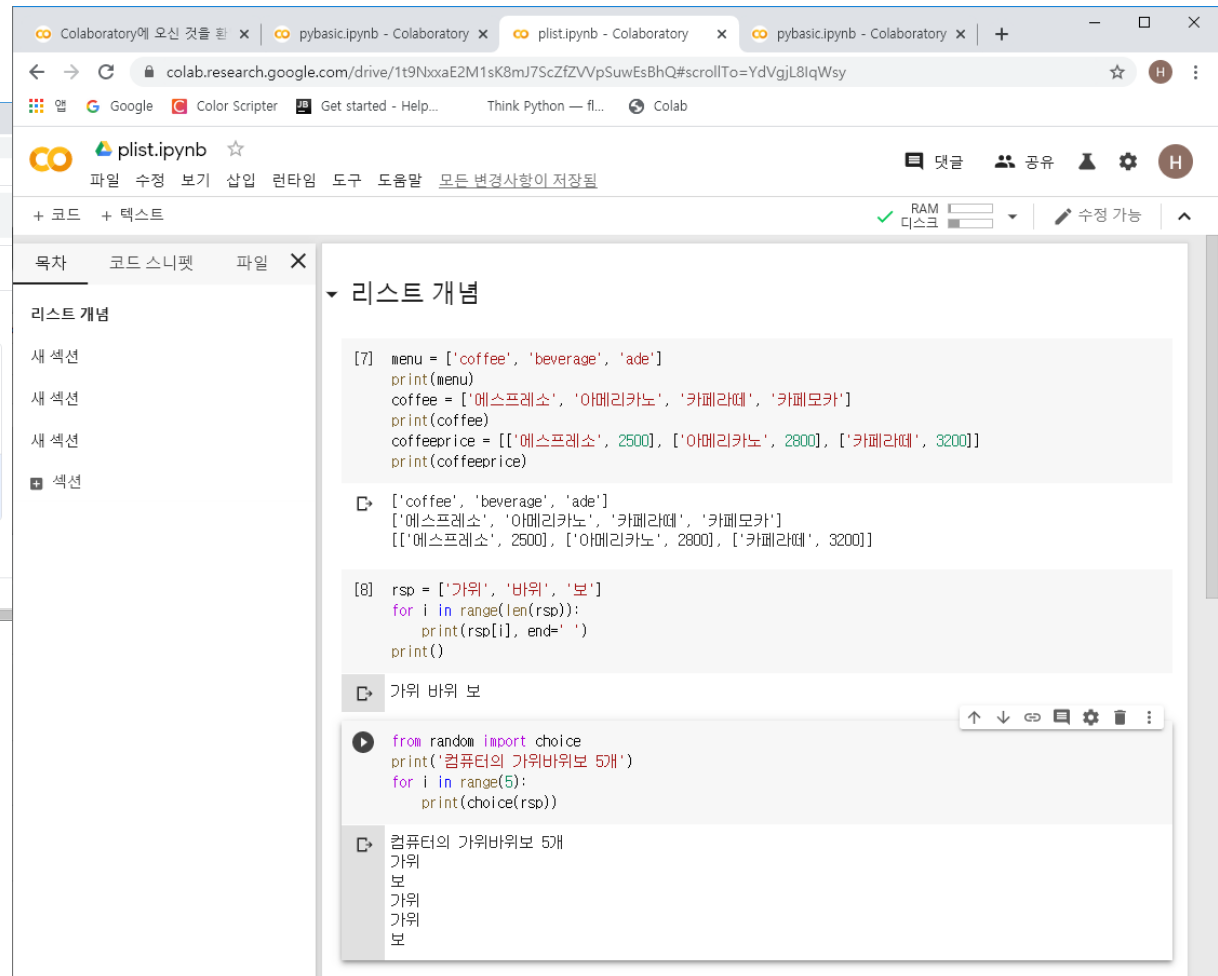
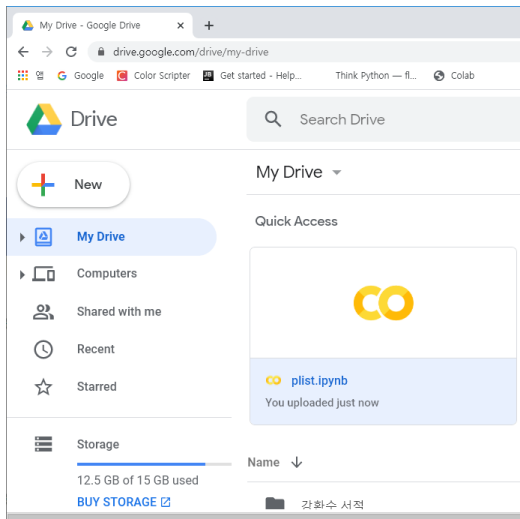

Colab Text Cell

- Text cell
 - 여러 줄 주석의 효과적인 시각화
 - 마크다운(Markdown) 문법
 - 자동 목차 생성



저장된 노트는 드라이브에서도 확인 가능

- 자신 구글 드라이브에 저장
 - Colab Notebooks



단축키

• 메뉴 도구 / 단축키

- 대부분의 작업은 단축키로 실행 가능
- 단축키 설정 가능
- 단축키 설정 화면 - Ctrl+M H

• 유용한 단축키

- 코드 셀 생성
 - Ctrl+M A(B)
- 코드 셀 실행
 - Ctrl+Enter
- 셀 삭제
 - Ctrl+M D
- 실행중인 셀 중단
 - Ctrl+M I
- 런타임 다시 시작
 - Ctrl+M .
- 코드(텍스트) 셀로 변환
 - Ctrl+M Y(M)
- 마지막 셀 작업 실행 취소
 - Ctrl+Shift+Z

키보드 환경설정

편집기 키 바인딩
default

☒ Enter 키를 눌러 제안 수락

단축키

단축키를 추가하거나 변경하려면 키 조합을 클릭한 다음 새 키를 입력하세요. Ctrl+M을(를) 다중 키-이벤트 단축키의 접두어로 사용할 수 있습니다.

| | | | |
|--------|------------------------------|------------------|------------|
| 단축키 설정 | .ipynb 다운로드 | 단축키 설정 | 선택한 출력 지우기 |
| 단축키 설정 | .py 다운로드 | 단축키 설정 | 선택항목 복사 |
| 단축키 설정 | Colaboratory FAQ 열기 | Ctrl+Shift+Enter | 선택항목 실행 |
| 단축키 설정 | GitHub Gist로 사본 저장 | 단축키 설정 | 선택항목 잘라내기 |
| 단축키 설정 | GitHub에 사본 저장 | 단축키 설정 | 설정 열기 |
| 단축키 설정 | Google 드라이브의 노트 별표표시/별표표시 해제 | 단축키 설정 | 세션 관리 |
| 단축키 설정 | Stack Overflow에 질문하기 | 단축키 설정 | 섹션 제목 셀 추가 |
| 단축키 설정 | 노트 공유 | Ctrl+Click | 셀 선택 토클 |
| 단축키 설정 | 이 문서 열기 | Ctrl+Shift+Y | 셀 작업 다시 실행 |

기본값 복원

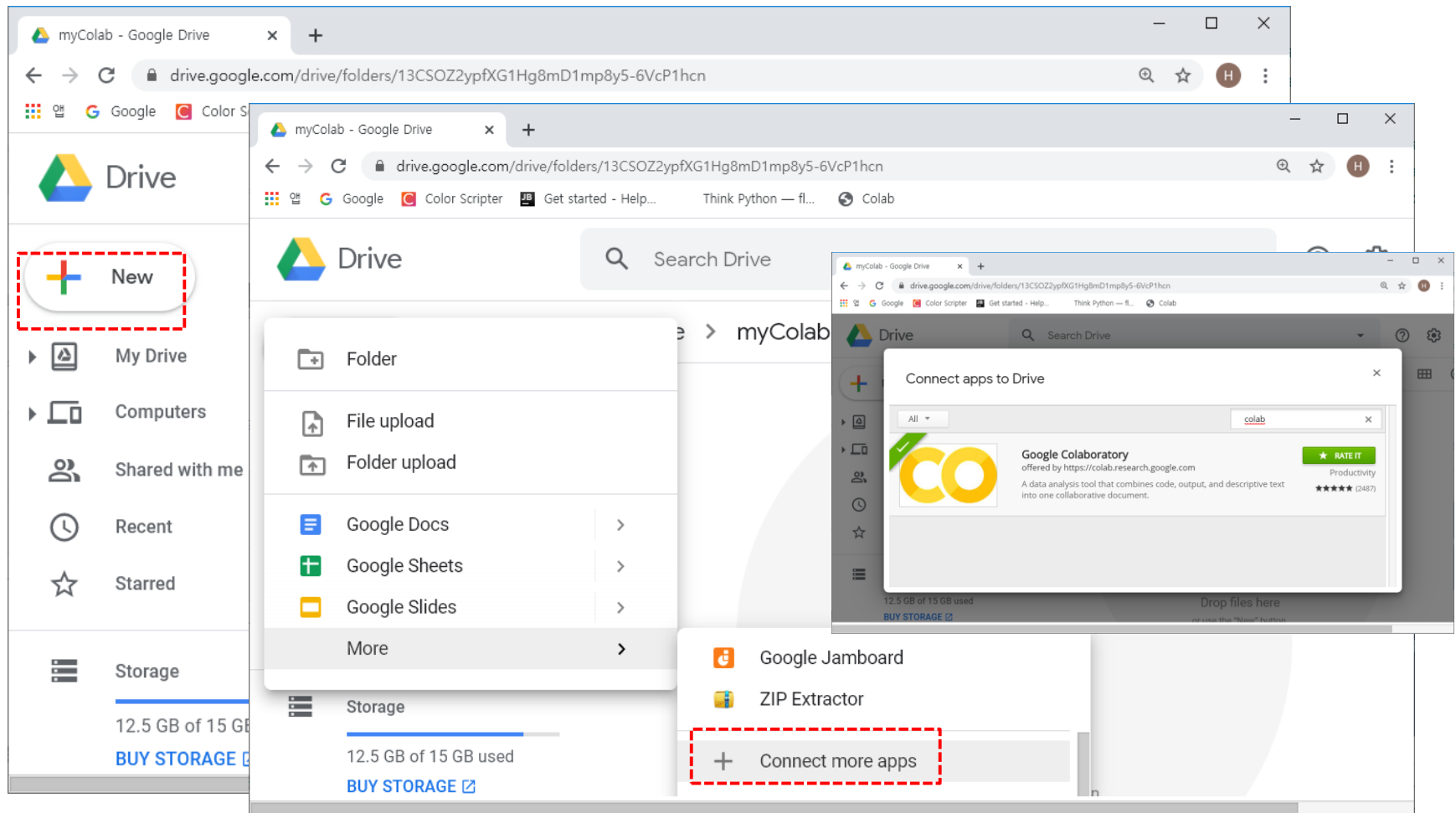
취소

저장

구글 드라이브에서 노트 파일 생성과 열기

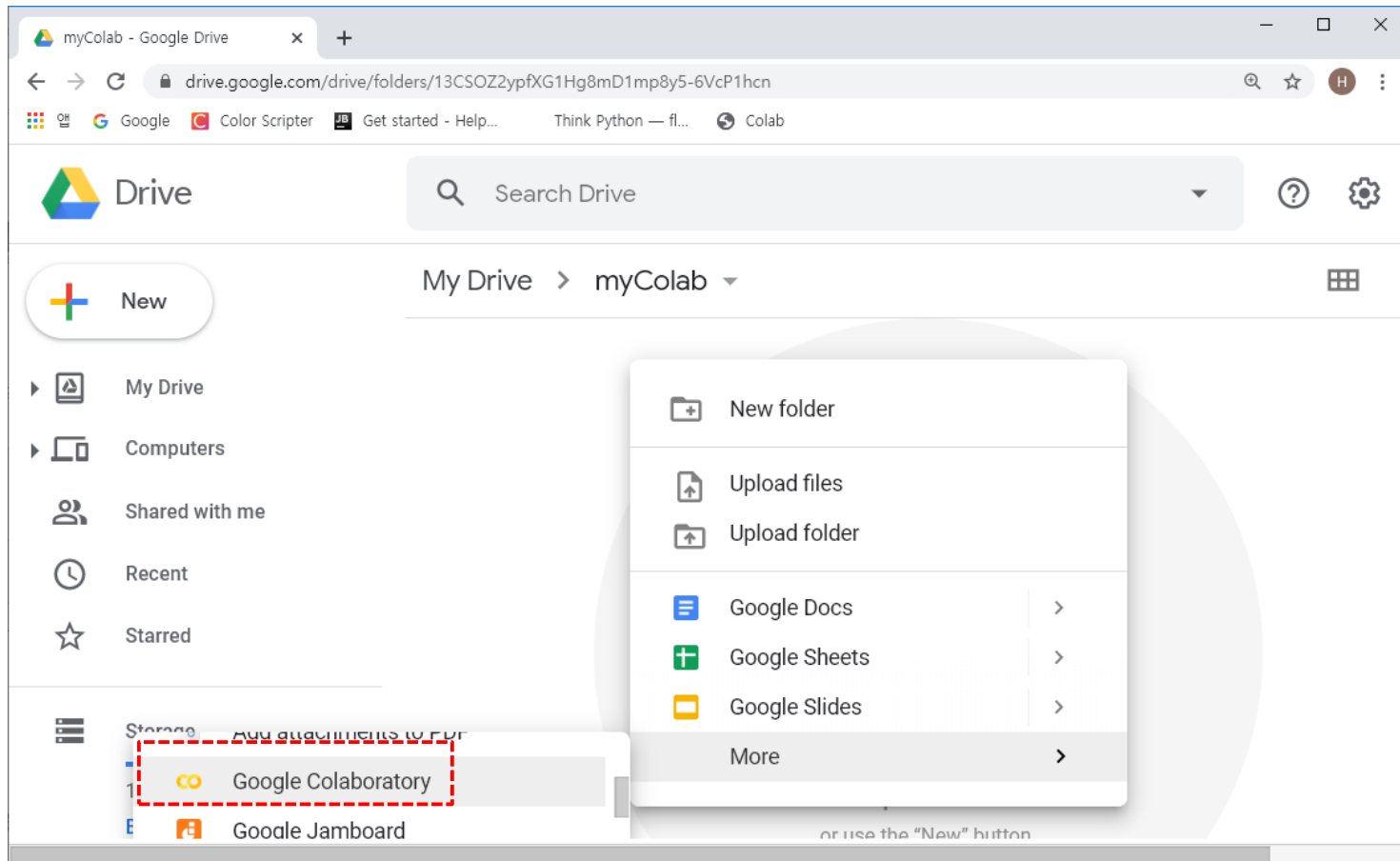
필요하면 특정 폴더 생성 한 후

- myColab



바탕에서도 가능

- 오른쪽 팝업 메뉴에서 Colab 선택 선택



파일 코딩 후 드라이브 확인

• 폴더 myColab

The screenshot shows a web browser with multiple tabs. The active tab is 'pyrand.ipynb - Colaboratory'. The notebook interface includes a toolbar with icons for file management, sharing, and settings. The code cell contains the following Python code:

```
[1] import random as rd

print(rd.random())
print(rd.randint(1, 10))
print(rd.randrange(1, 10))
print(rd.randrange(10))
print(rd.choice(range(-10, 10)))
```

The output of the first cell is displayed below the code:

```
<built-in method random of Random object at 0x2212a18>
4
9
2
-9
```

The second cell contains the code:

```
[2] print(rd.random()) # [0, 1)
```

The output is:

```
0.6175009419284109
```

The third cell contains the code:

```
[7] print(rd.uniform(5, 10))
```

The output is:

```
6.379668203736781
```

The fourth cell contains the code:

```
[11] print(rd.choices(range(3), k=5))
```

The output is:

```
[2, 0, 2, 1, 1]
```

The fifth cell contains the code:

```
print(rd.choice(range(100)))
```

The output is:

```
34
```

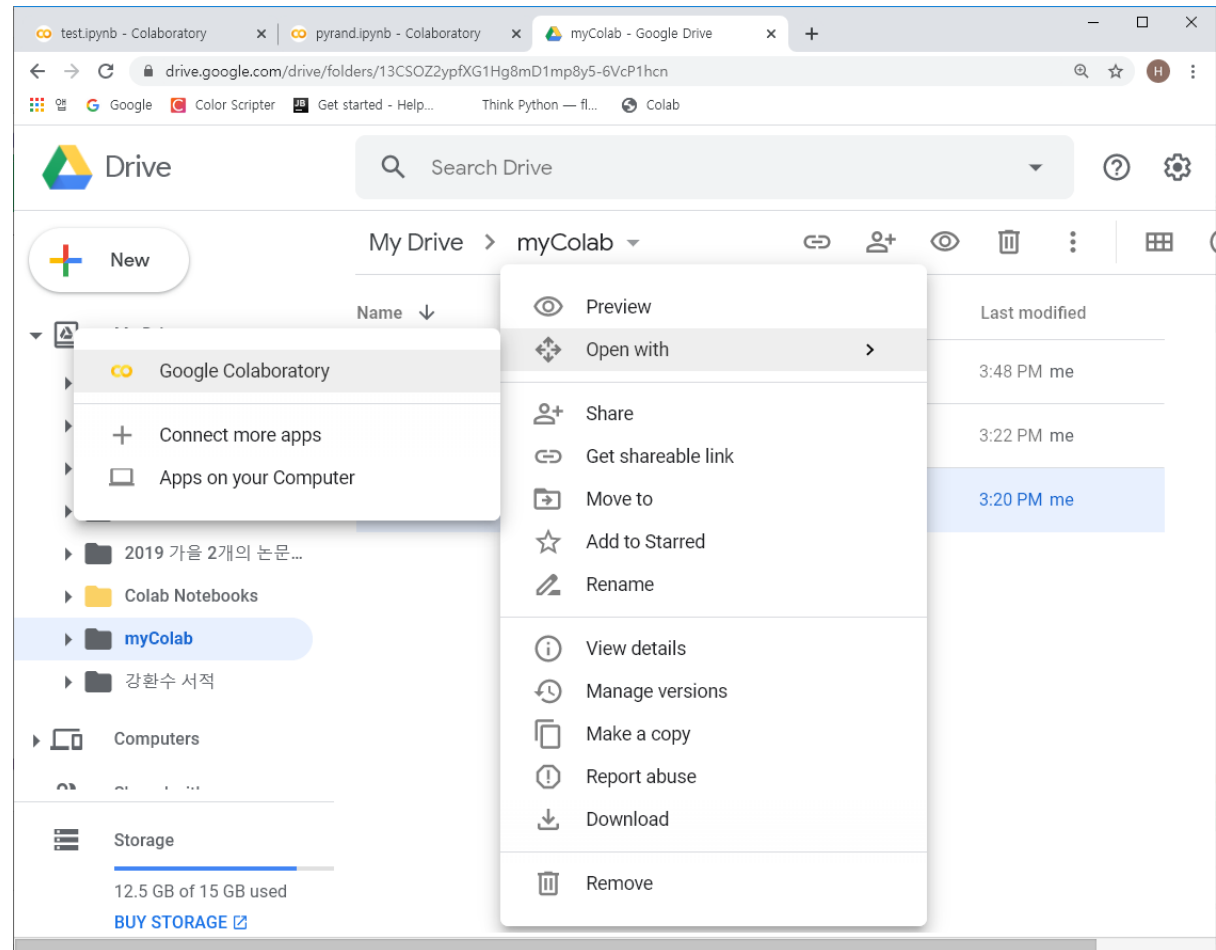
12.5 GB of 15 GB used
[BUY STORAGE](#)

The screenshot shows the Google Drive interface. The 'My Drive' section is expanded, showing a folder named 'myColab'. Inside this folder, there are two files: 'test.ipynb' and 'pyrand.ipynb'. The table below summarizes the files shown in the interface.

| Name | Owner | Last modified |
|--------------|-------|---------------|
| test.ipynb | me | 3:22 PM me |
| pyrand.ipynb | me | 3:20 PM me |

노트 파일 열기

- 파일 선택 후
 - 오른쪽 팝업 메뉴



자신 PC 파일 열기

메뉴 선택

- 파일 / 노트 업로드
- 또는 다음 대화상자에서 가장 오른쪽 '업로드' 선택
 - 파일 선택
 - 또는 드래그 & 드롭



파일 업로드 후 수정

Colab Notebooks - Google Drive x Chapter1.ipynb - Colaboratory x +

colab.research.google.com/drive/1uE30ykUwaQ0NEjHvphecWm4I4QZqnFTP

Chapter1.ipynb ☆

파일 수정 보기 삽입 런타임 도구 도움말 오후 6:11에 마지막으로 저장됨

+ 코드 + 텍스트

Python for Data Analysis

1장 - 시작하기 전에

```
#
pandas : panel data, python data analysis

~$ python
Python 2.7.5 (default, Sep 12 2013, 21:33:34)
[GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.0.68)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>

~$ ipython --pylab
Python 2.7.5 (default, Sep 12 2013, 21:33:34)
Type "copyright", "credits" or "license" for more information.

IPython 1.2.1 -- An enhanced Interactive Python.
?      -> Introduction and overview of IPython's features.
Quickref -> Quick reference.
help    -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
```

Colab Notebooks - Google Drive x Chapter1.ipynb - Colaboratory x +

colab.research.google.com/drive/1uE30ykUwaQ0NEjHvphecWm4I4QZqnFTP#scrollTo=sv0trpDUXvCi

Chapter1.ipynb ☆

파일 수정 보기 삽입 런타임 도구 도움말 모든 변경사항이 저장됨

+ 코드 + 텍스트

RAM 디스크 수정 가능

```
$ !python notebook --pylab=inline

[3] import pandas

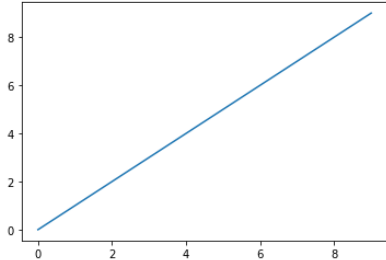
[4] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

[7] plt.plot(range(10))

[<matplotlib.lines.Line2D at 0x7f6414f7f240>]

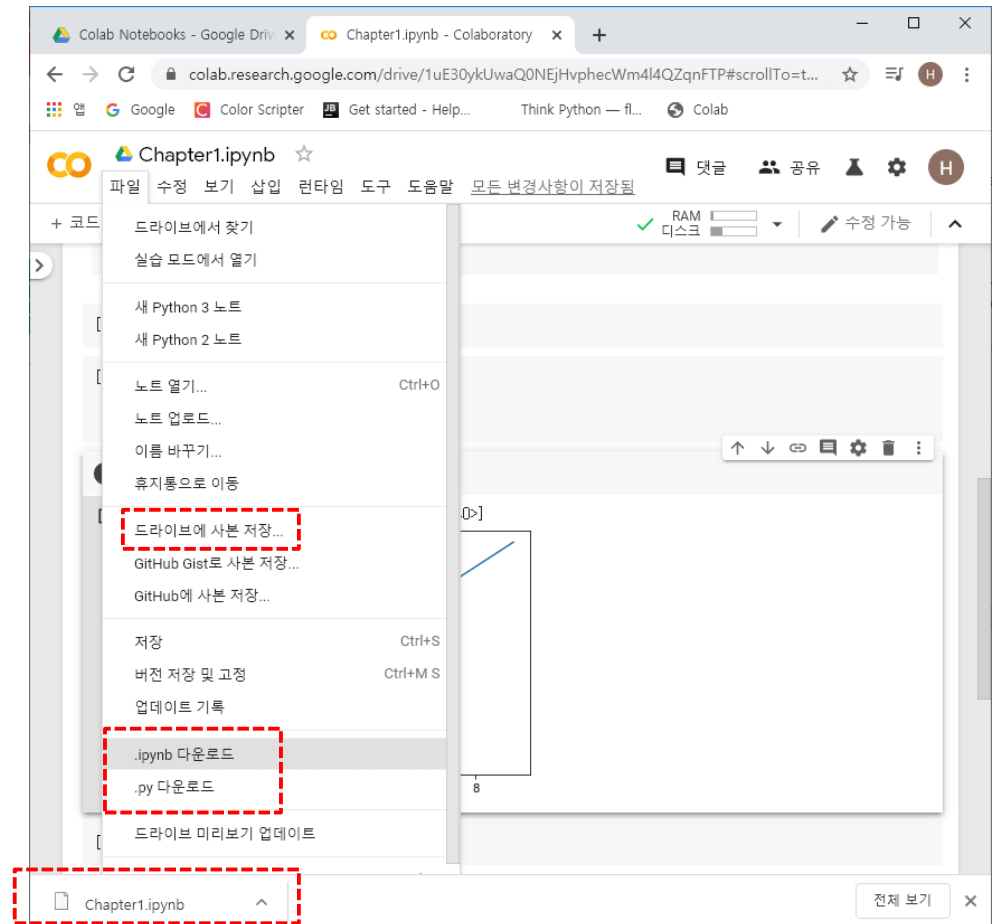
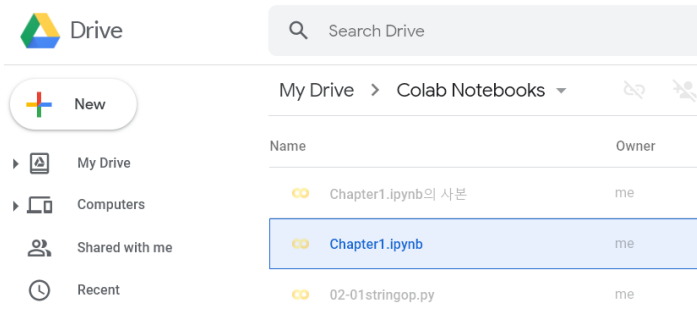
[8] print('헬로 월드!')

헬로 월드!
```



내 PC에 다시 저장

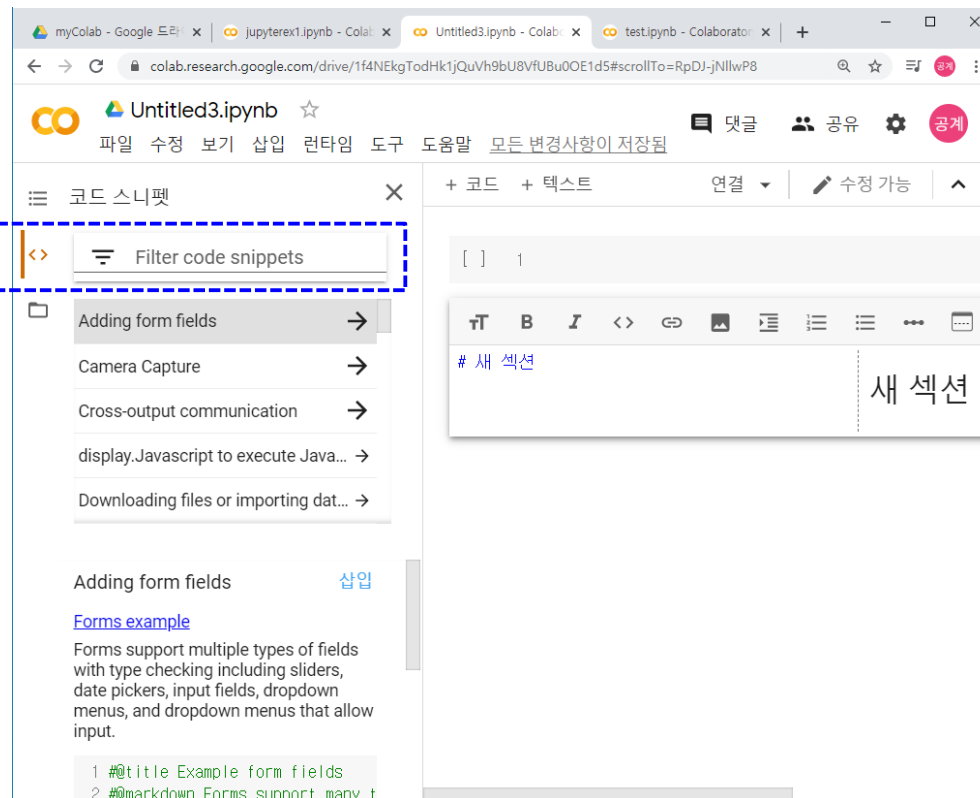
- **.ipynb 다운로드**
 - 일반 파이썬 소스로도 가능
 - .py 다운로드
- **드라이브에 사본 저장...**
 - 드라이브의 다음 폴더에 저장
 - **Colab Notebooks**



구글 Colab 샘플 코드
코드 스니펫

코드 스니펫

- 파일 생성에서
 - 화면 왼쪽 펼쳐
 - 코드 스니펫(snippets) 선택
 - 매우 다양한 부분 코드



자신의 파일에 스니펫을 삽입

- 자신의 드라이브에서 직접 노트북 파일 생성 한 후
 - /myColab/gg sample/code snippets sample.ipynb
- 왼쪽 확장
 - 제목과 소스로 구성
 - 삽입을 통해 오른쪽과 같이 구성

The screenshot shows a Google Colab notebook titled 'Untitled0.ipynb'. On the left, a sidebar displays a list of code snippets under the heading 'Filter code snippets'. The snippets include:

- Using BigQuery with Pandas API
- Visualization: Bar Plot in Altair
- Visualization: Histogram in Altair
- Visualization: Interactive Brushing in Altair
- Visualization: Interactive Scatter Plot in ...
- Visualization: Linked Brushing in Altair

The 'Visualization: Interactive Scatter Plot in ...' snippet is selected, showing its content:

```
# load an example dataset
from vega_datasets import data
cars = data.cars()

# plot the dataset, referencing dataframe column names
import altair as alt
alt.Chart(cars).mark_point().encode(
    x='Horsepower',
    y='Miles_per_Gallon',
    color='Origin',
).interactive()
```

On the right, the notebook content shows two sections:

- Visualization: Bar Plot in Altair**

```
# load an example dataset
from vega_datasets import data
cars = data.cars()

# plot the dataset, referencing dataframe column names
import altair as alt
alt.Chart(cars).mark_bar().encode(
    x='mean(Miles_per_Gallon)',
    y='Origin',
    color='Origin'
)
```
- Visualization: Interactive Scatter Plot in Altair**

```
[ ] # load an example dataset
from vega_datasets import data
cars = data.cars()

# plot the dataset, referencing dataframe column names
import altair as alt
alt.Chart(cars).mark_point().encode(
    x='Horsepower',
    y='Miles_per_Gallon',
    color='Origin'
).interactive()
```

파일 실행

- 실행 후
- 저장

code snippets sample.ipynb ☆

파일 수정 보기 삽입 런타임 도구 도움말

+ 코드 + 텍스트

목록 코드 스니펫 파일

Filter code snippets

- Using BigQuery with Pandas API →
- Visualization: Bar Plot in Altair →
- Visualization: Histogram in Altair →
- Visualization: Interactive Brushing in Altair →
- Visualization: Interactive Scatter Plot in ... →
- Visualization: Linked Brushing in Altair →

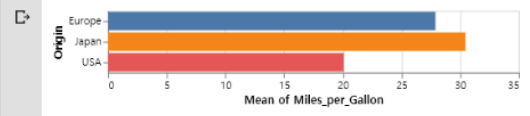
Visualization: Interactive Scatter Plot...

Altair lets you easily create an interactive scatter

Visualization: Bar Plot in Altair

```
# load an example dataset
from vega_datasets import data
cars = data.cars()

# plot the dataset, referencing dataframe column names
import altair as alt
alt.Chart(cars).mark_bar().encode(
    x='mean(Miles_per_Gallon)',
    y='Origin',
    color='Origin'
)
```



My Drive > myColab > gg sample

New

code snippets sample.ipynb

myColab

gg sample

| Name | Owner |
|----------------------------|-------|
| code snippets sample.ipynb | me |

Visualization: Interactive Scatter Plot in Altair

