



# 인공신경망과딥러닝입문

## Lecture 03. 인공지능 개발 환경 구축하기

동덕여자대학교  
데이터사이언스 전공  
권 범

# 목차

- ❖ 01. 두 가지 프로그래밍 환경
- ❖ 02. Cloud 방식의 개발 환경 구축하기
- ❖ 03. Standalone 방식의 개발 환경 구축하기

# 01. 두 가지 프로그래밍 환경

02. Cloud 방식의 개발 환경 구축하기

03. Standalone 방식의 개발 환경 구축하기

# 01. 두 가지 프로그래밍 환경

## ❖ 두 가지 프로그래밍 환경

① Cloud 방식

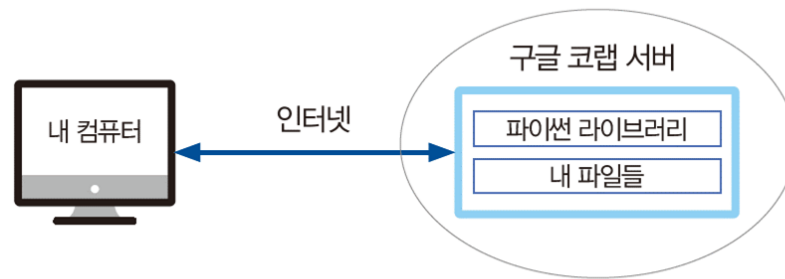
② Standalone 방식

정답은 없음.  
나에게 적절한 환경을 선택하면 됨

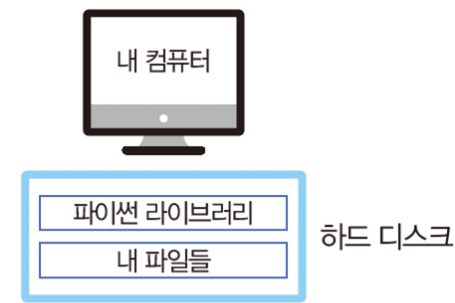
# 01. 두 가지 프로그래밍 환경

## ❖ Cloud 방식

- 프로그램과 데이터가 서버에 저장되고 관리됨
  - 서버에 환경이 대부분 갖추어져 있어 로그인만 하면 바로 프로그래밍 가능함
  - 인터넷에 연결만 되어 있으면 어느 곳에서나 개발 및 협업이 가능함
  - 한계: 내 프로젝트에 최적의 환경을 갖추는 없음
- ① Google: Colab(Colaboratory, 코랩)
  - ② Amazon: SageMaker(세이지메이커)
  - ③ Microsoft: Azure(애저)



(a) 클라우드 방식

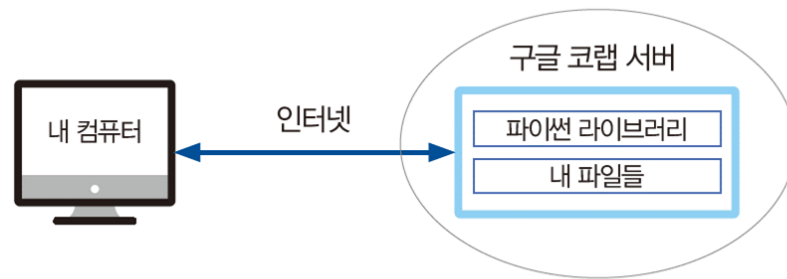


(b) 스탠드얼론 방식

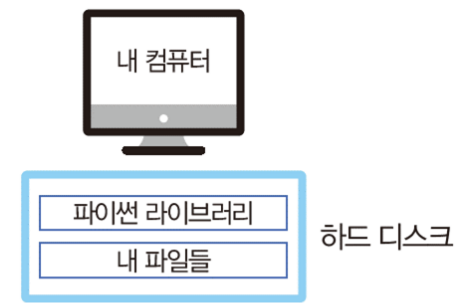
# 01. 두 가지 프로그래밍 환경

## ❖ Standalone 방식

- 프로그램과 데이터가 자신의 컴퓨터에 저장됨
- 소프트웨어를 설치하고 환경을 스스로 구축해야 함
- 장점: 자신의 최적인 환경 구축이 가능함



(a) 클라우드 방식



(b) 스탠드얼론 방식

# 01. 두 가지 프로그래밍 환경

## ❖ 프로그래밍 환경에 대한 좋은 태도 (1/2)

최근 여러 가지 프로그래밍 환경이 주어져 있음



인공지능 SW 프로그래머 층이 두터워졌다는 뜻으로 해석 할 수 있음

# 01. 두 가지 프로그래밍 환경

## ❖ 프로그래밍 환경에 대한 좋은 태도 (2/2)

시범적으로 프로그래밍을 해보려는 사람.  
본격적으로 제품을 개발하는 사람.  
이 두 사람의 요구사항은 다를 수밖에 없음

따라서 다양한 프로그래밍 환경이  
저마다의 목적으로 개발되고 있는 것임

두 가지 프로그래밍 환경 중에 반드시 하나를 선택해야 하는 것이 아니며,  
두 환경 사이를 쉽게 전환할 수 있는 수준의 익숙함을 확보하는 것이  
현명한 태도라고 볼 수 있겠음



## 02. Cloud 방식의 개발 환경 구축하기

01. 두 가지 프로그래밍 환경

03. Standalone 방식의 개발 환경 구축하기

## 02. Cloud 방식의 개발 환경 구축하기

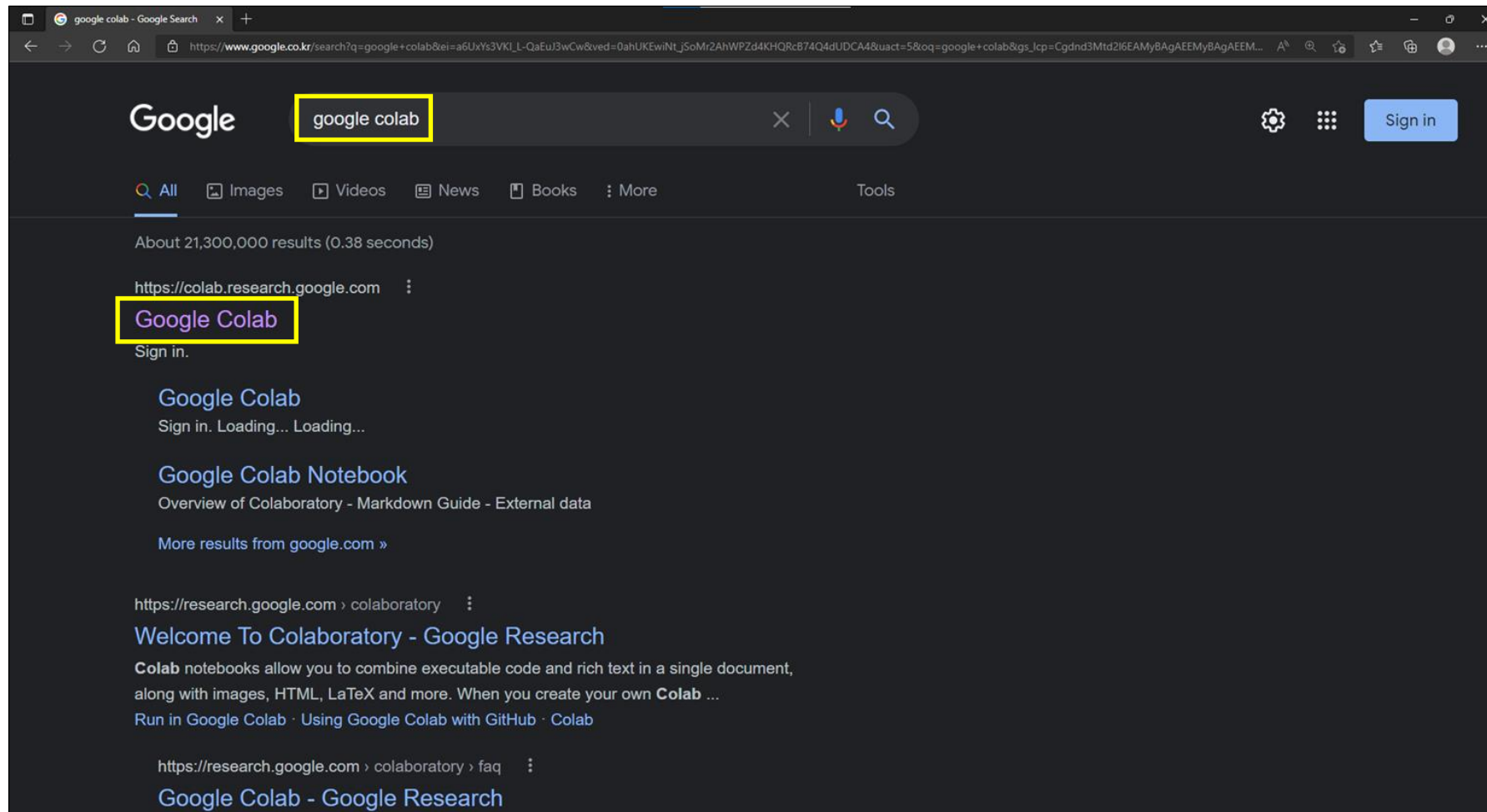
### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (1/13)

- ① Google Colab 사이트(<https://colab.research.google.com>)에 접속
- ② Google 계정으로 로그인(계정이 없다면, 만든 다음에 로그인)
- ③ 파이썬 프로그래밍을 함
- ④ 프로그래밍한 프로그램 파일은 자동으로 구글 드라이브에 저장됨

## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (2/13)

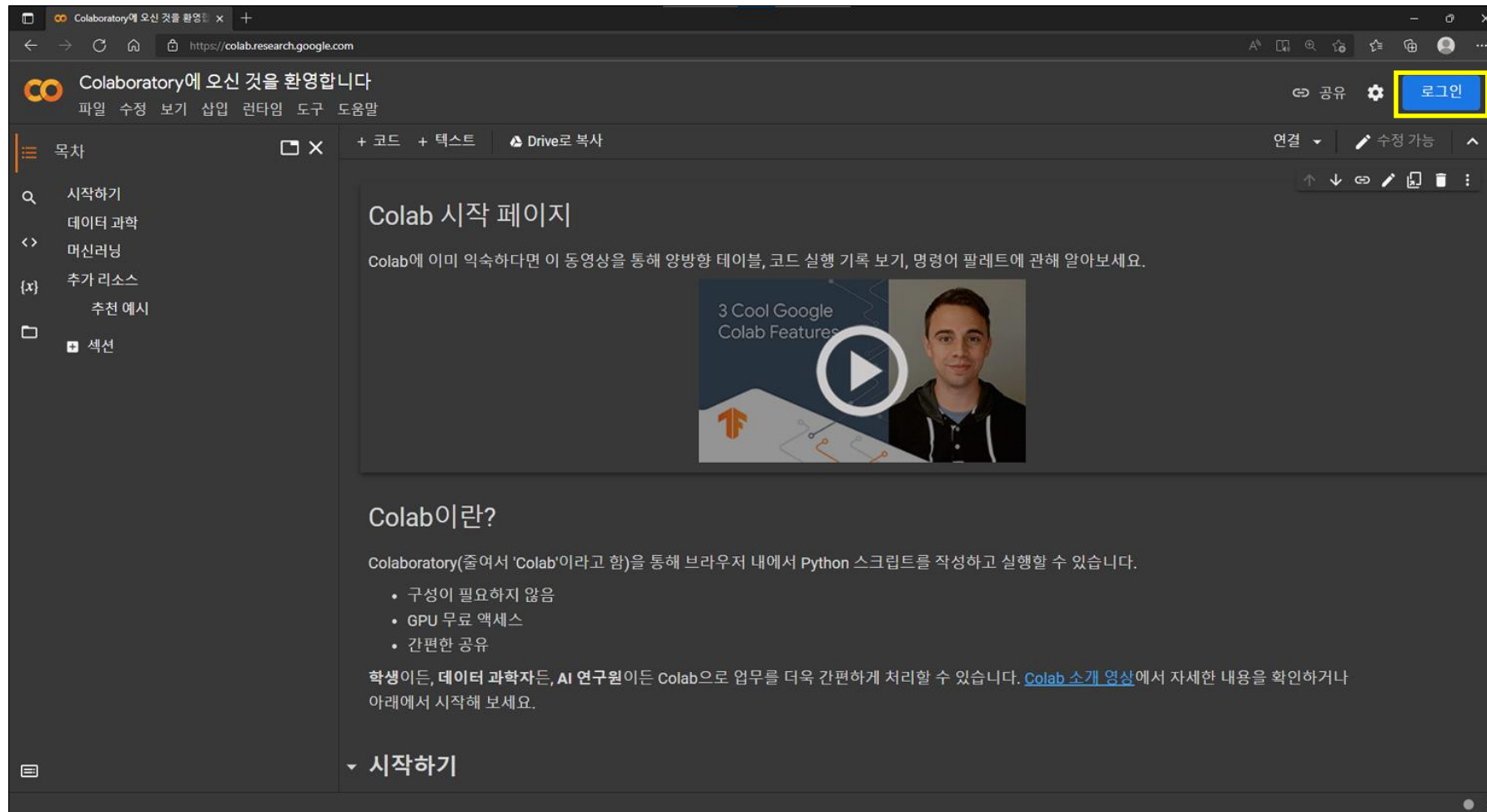
- 웹 브라우저(Web Browser)에서 Google Colab을 검색하고, 클릭



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (3/13)

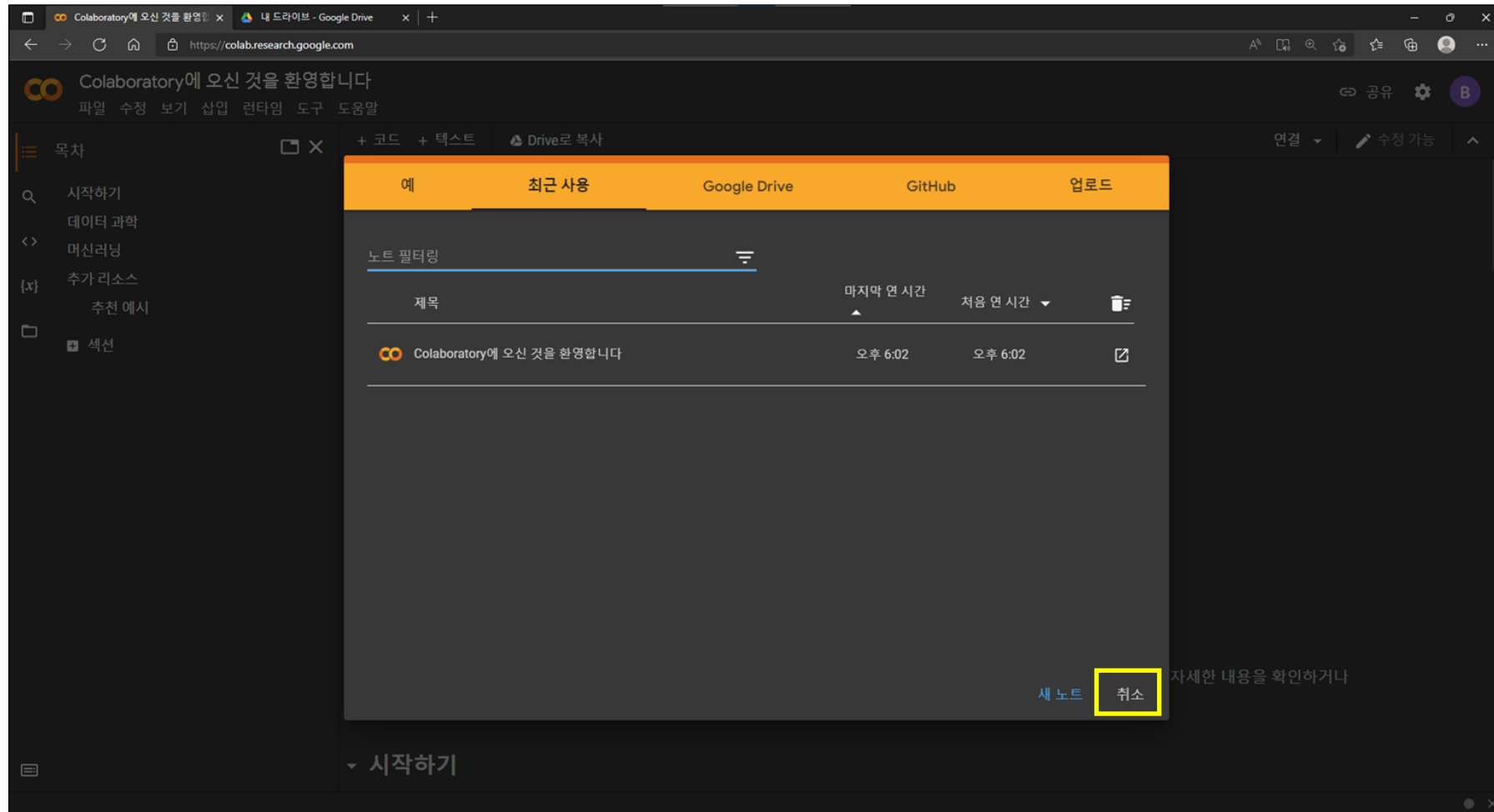
- 본인의 Google 계정으로 로그인



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (4/13)

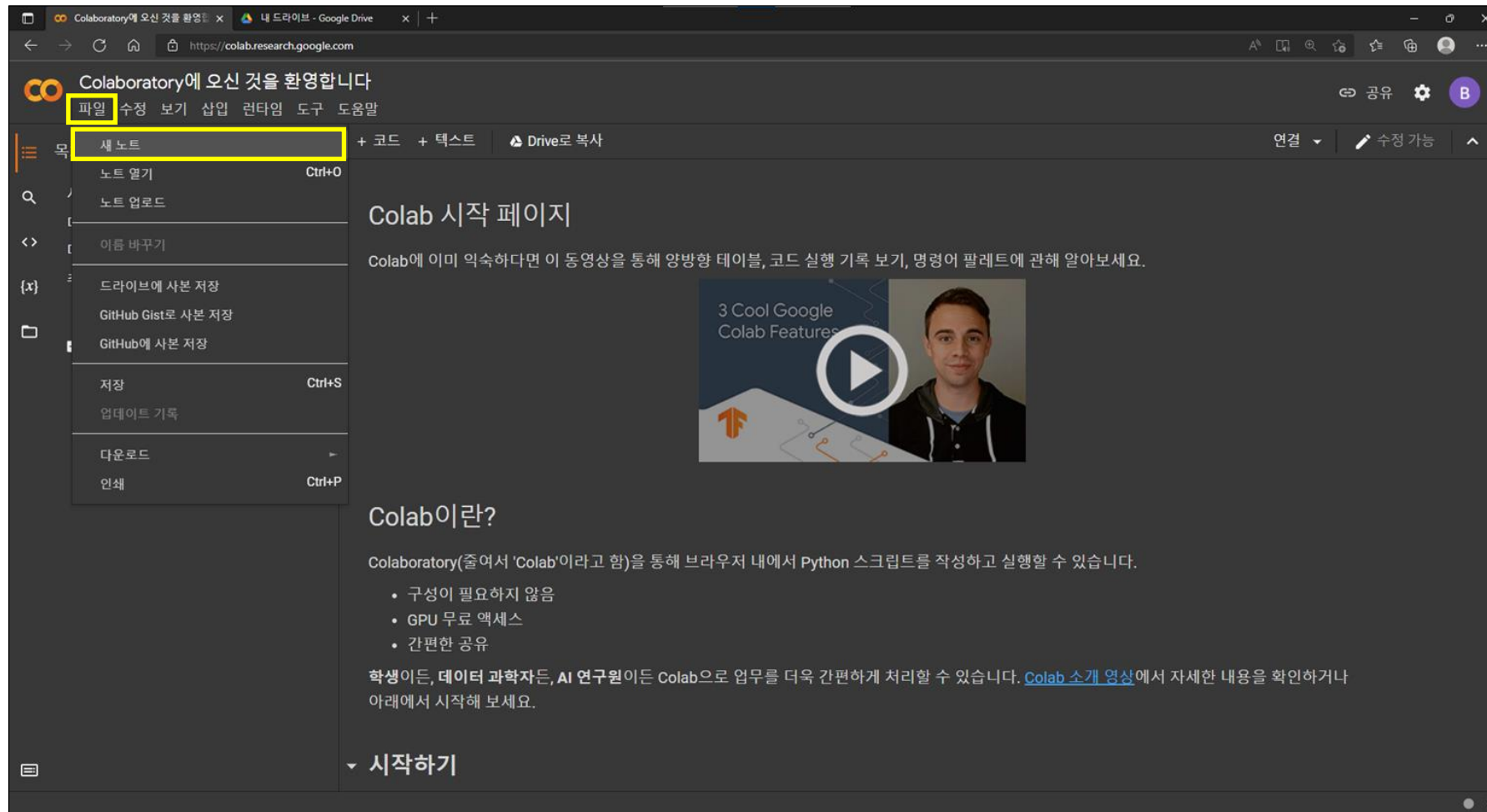
- 취소 버튼을 눌러 팝업 창을 닫아 줌



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (5/13)

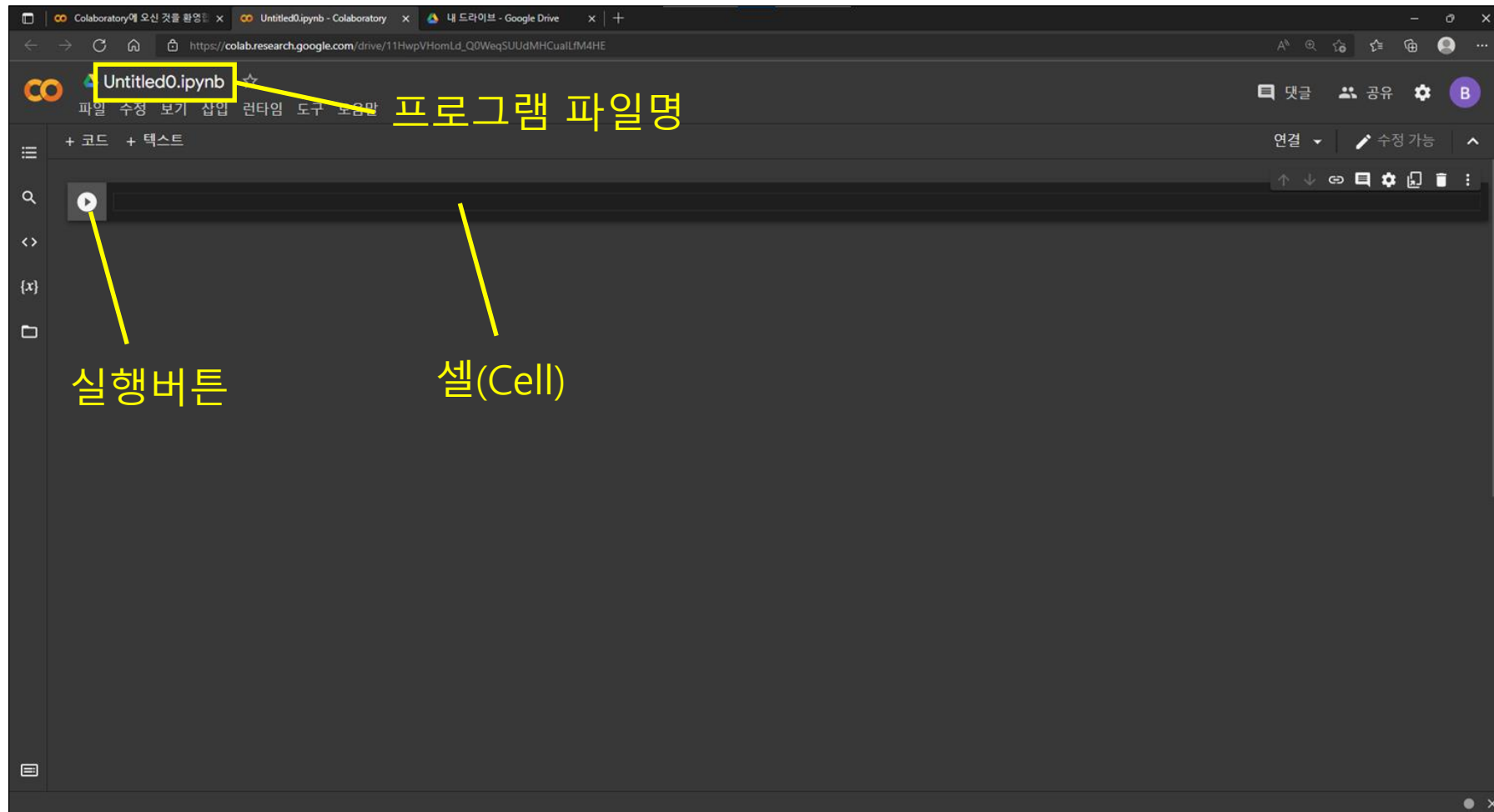
- [파일] → [새 노트]를 클릭



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (6/13)

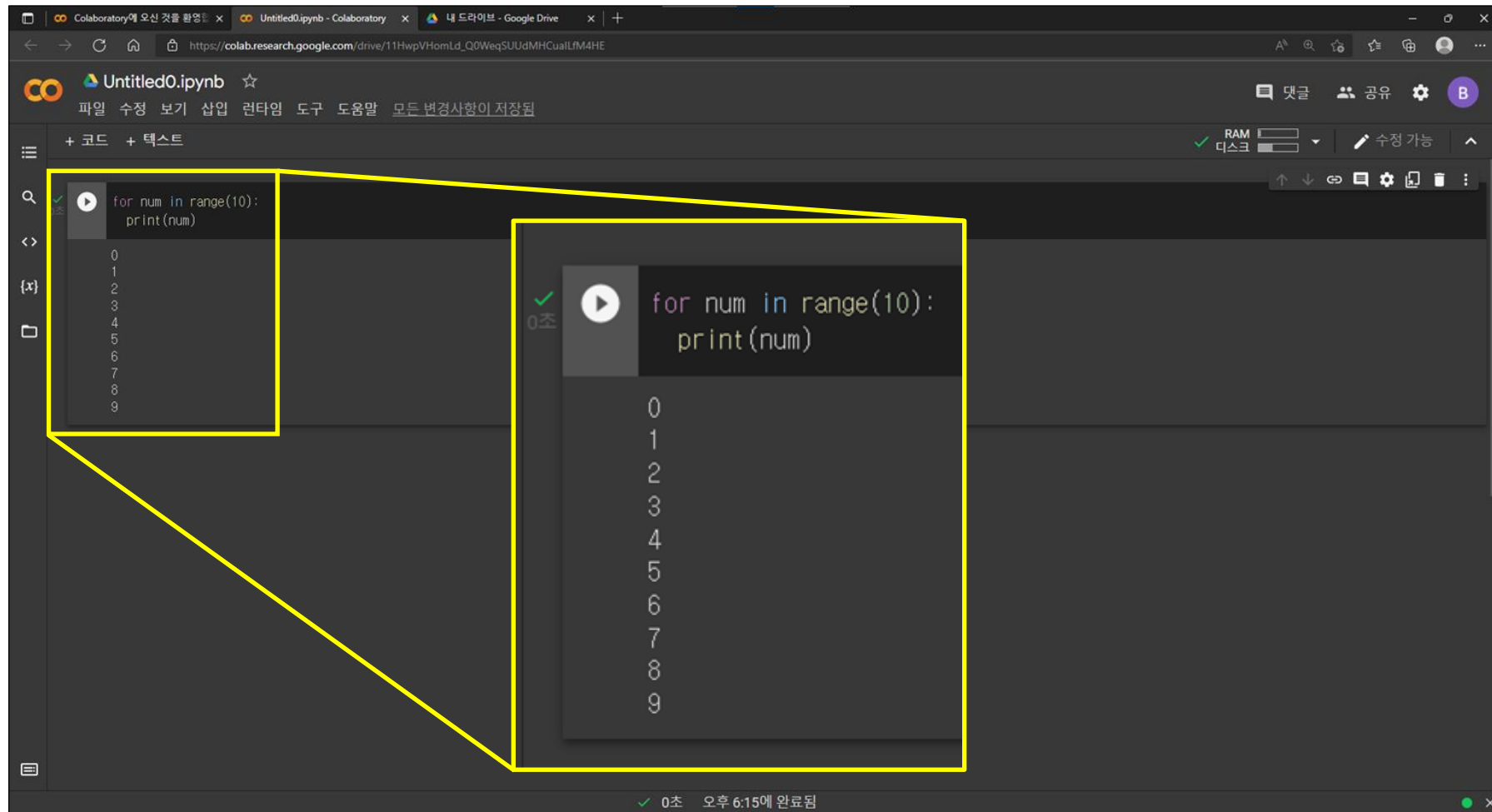
- 생성된 .ipynb 파일을 확인할 수 있고, 여기서 ipynb은 Interactive Python Notebook의 약어임



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (7/13)

- 파이썬 코드를 입력하고 실행시켜 결과를 확인

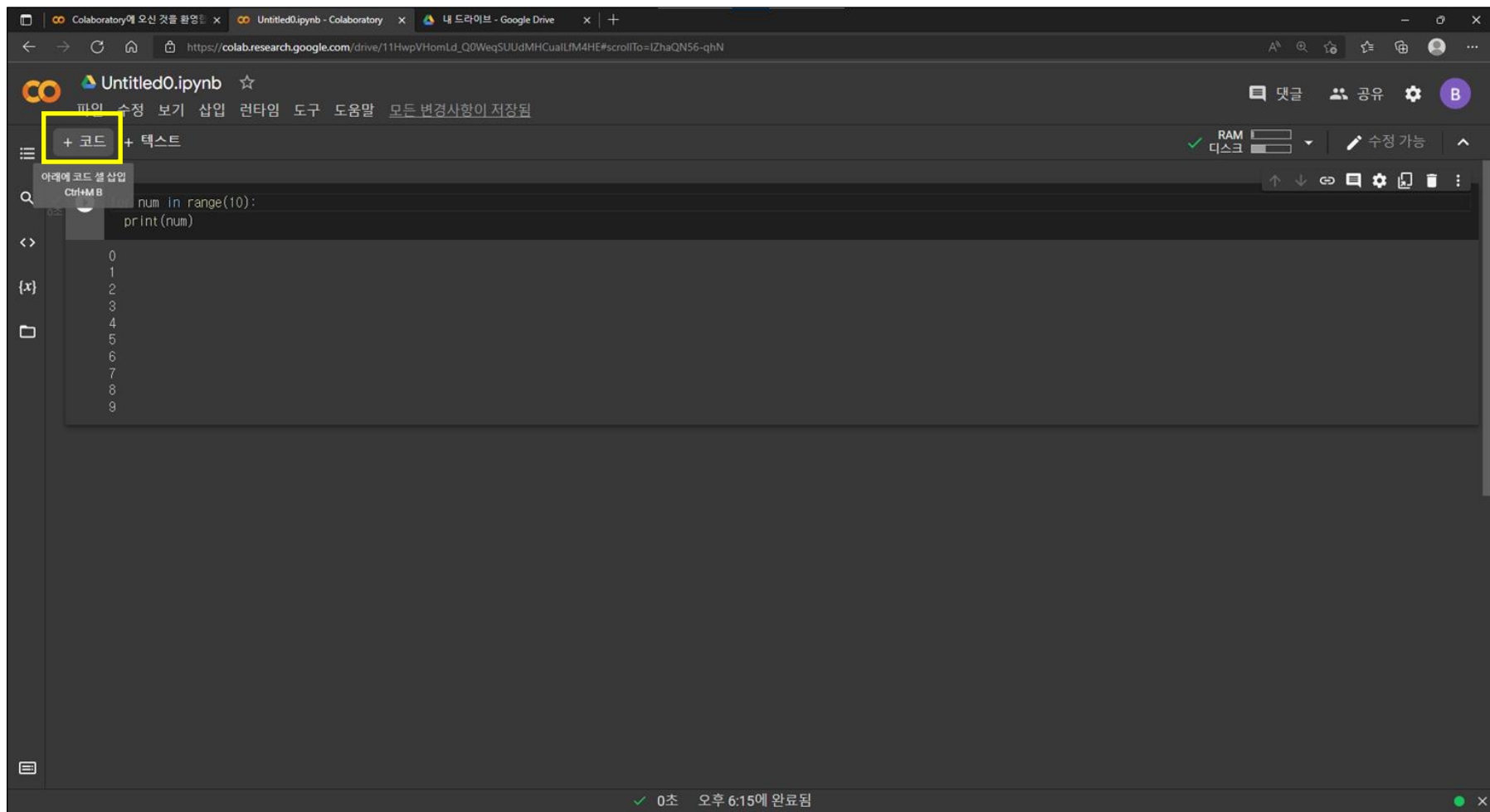




## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (8/13)

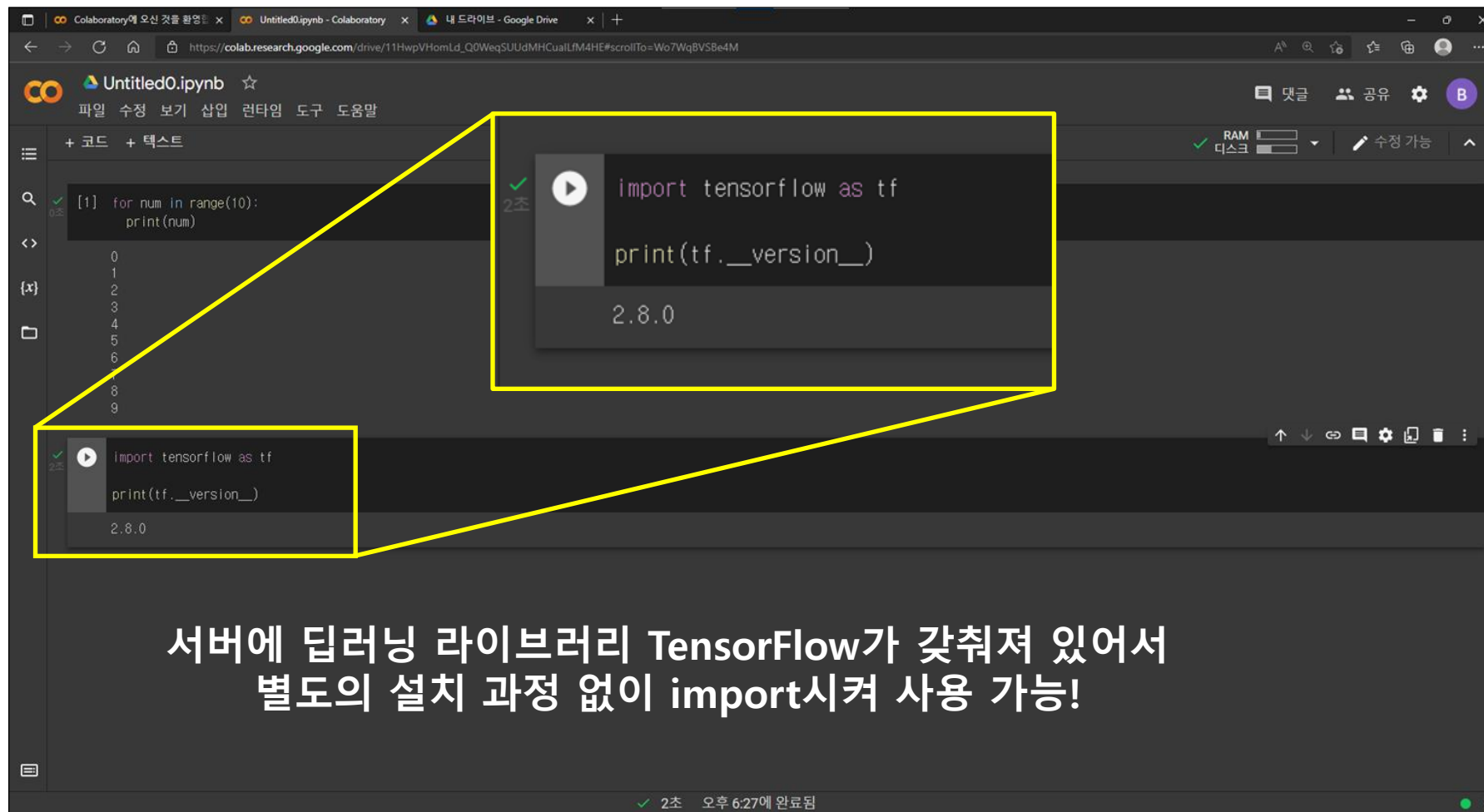
- 코드 작성을 위한 셀을 추가하기 위해서, [+ 코드] 버튼을 클릭(단축키: Ctrl + M B)



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (9/13)

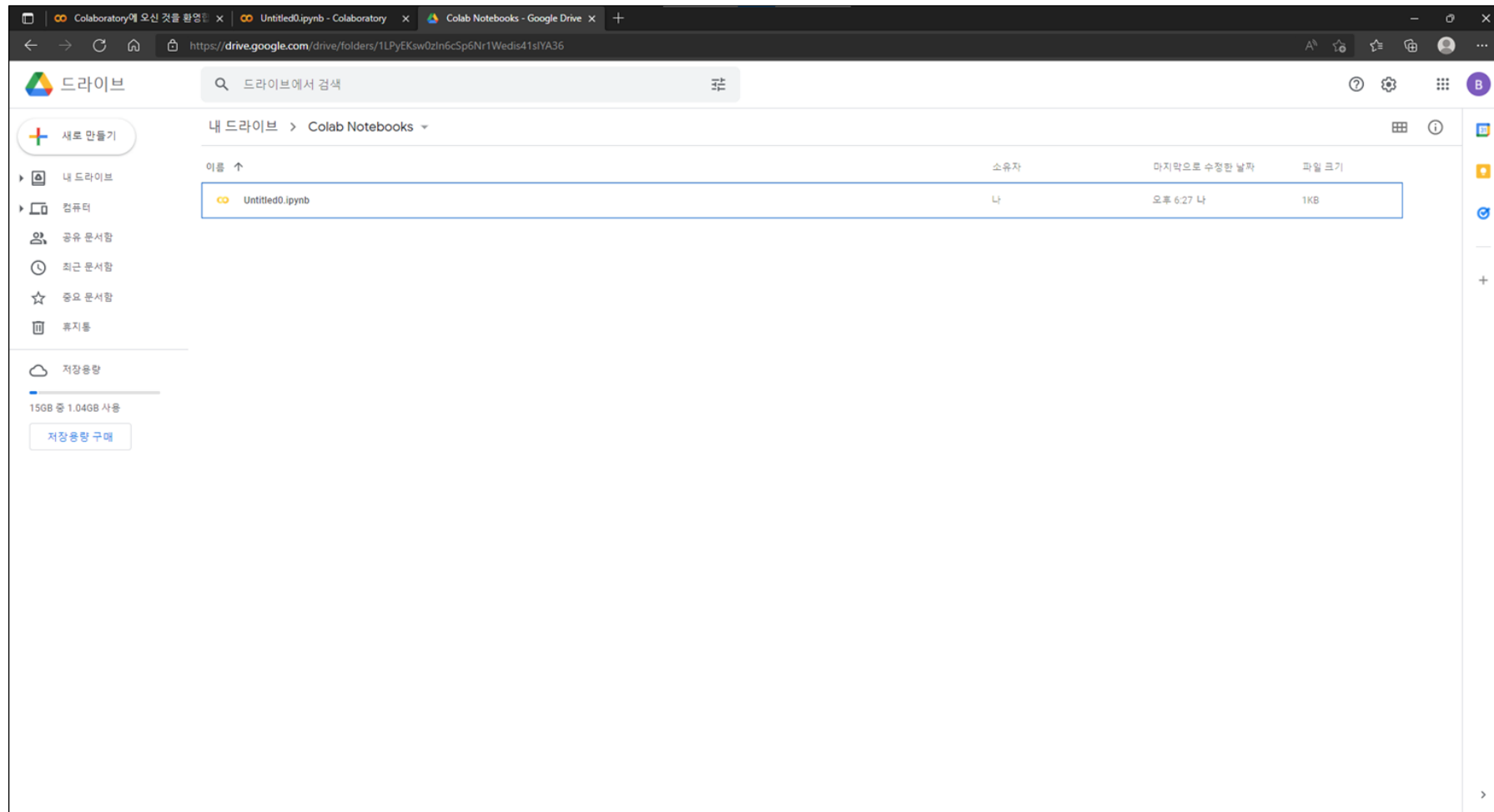
- 추가된 셀에 파이썬 코드를 입력하고 실행시켜 결과를 확인



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (10/13)

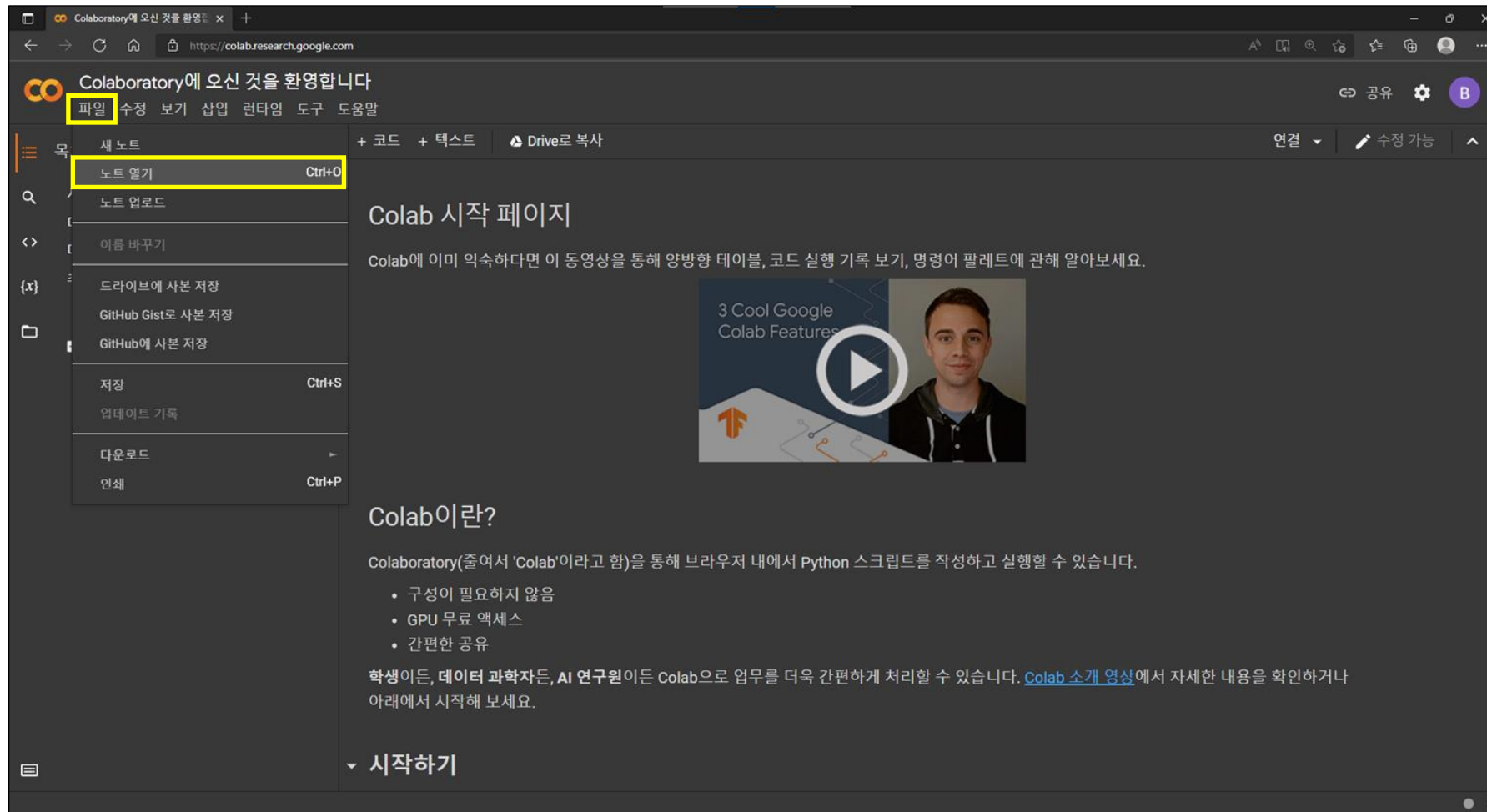
- 저장 및 작업한 .ipynb 파일은 Google 드라이브의 [내 드라이브] – [Colab Notebooks]에 저장됨



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (11/13)

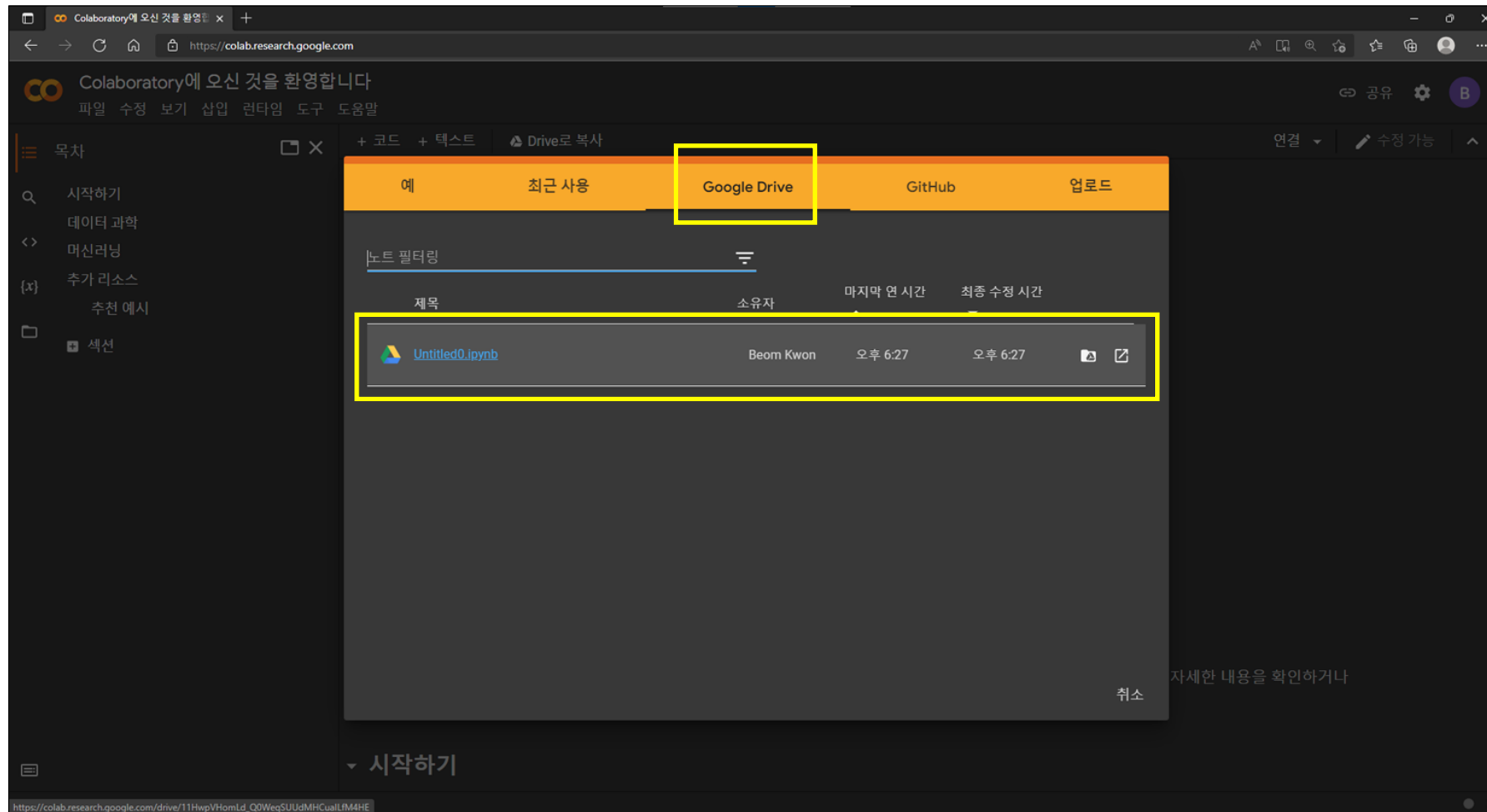
- 저장되어 있는 파일을 열기 위해서, [파일] – [노트 열기] 버튼을 클릭



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (12/13)

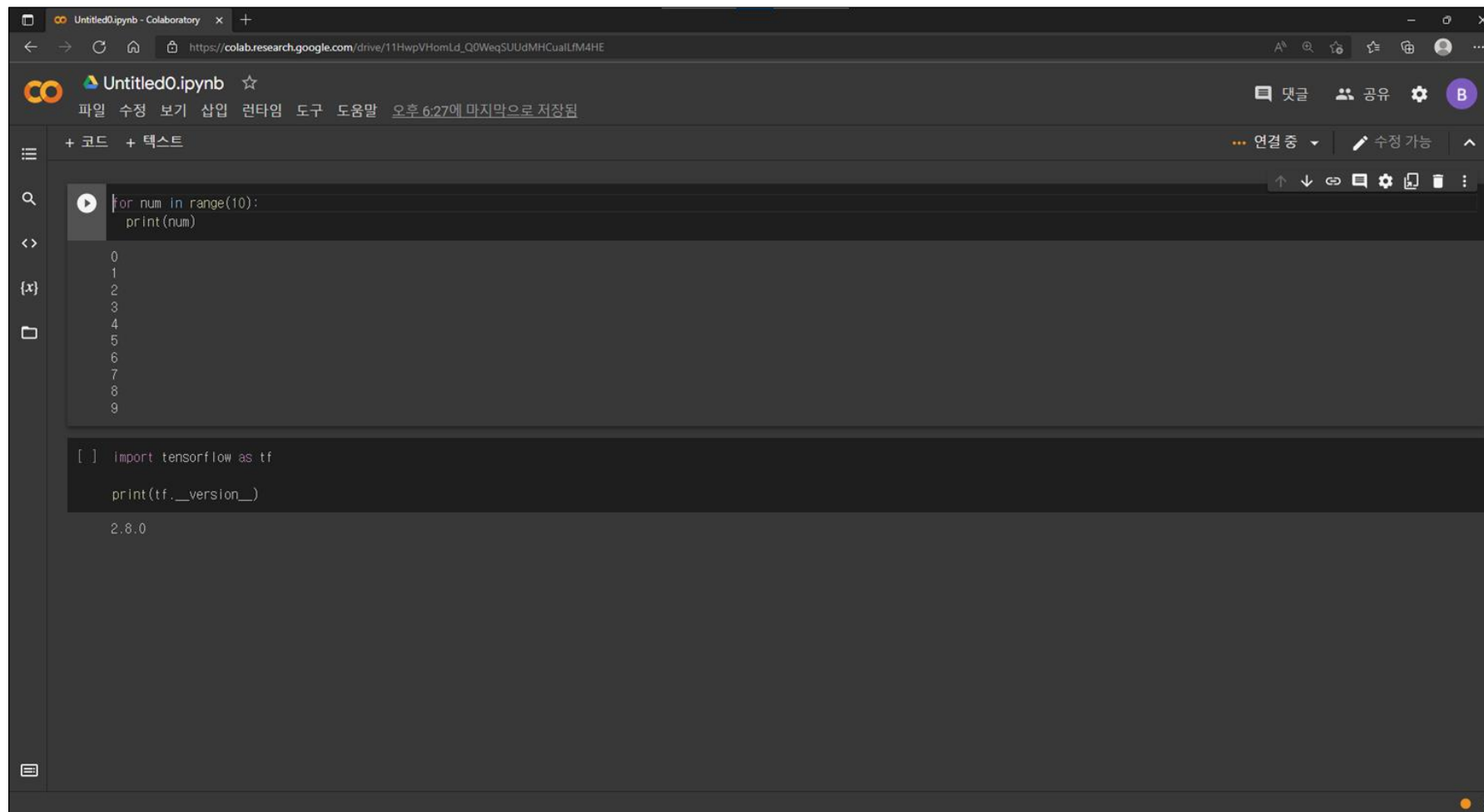
- 팝업 창에서 [Google Drive] 탭에서 열고자 하는 파일을 클릭



## 02. Cloud 방식의 개발 환경 구축하기

### ❖ 구글 코랩(Google Colab)을 사용하는 절차 (13/13)

- .ipynb 파일이 열린 것을 확인



## 03. Standalone 방식의 개발 환경 구축하기

- 01. 두 가지 프로그래밍 환경
- 02. Cloud 방식의 개발 환경 구축하기

## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (1/12)

- 파이썬은 무료로 다운로드할 수 있어 비용을 지불하지 않아도 사용에 제한이 없음
- 파이썬을 설치하는 방법은 크게 2가지가 있음
  - ① 파이썬 공식 홈페이지 → 파이썬 Installer 다운로드 → 파이썬 설치
  - ② 아나콘다(Anaconda) 공식 홈페이지 → 아나콘다 Installer 다운로드 → 아나콘다 설치

위의 두 가지 방법 중 아나콘다를 통한  
파이썬 설치를 권장함



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 설치 (2/12)

- 아나콘다를 통한 설치를 추천하는 이유는 다음과 같음

① 데이터 과학 및 딥러닝 관련된 수 많은 라이브러리를 따로 설치할 필요 없음

- ✓ 파이썬에서 어떠한 외부 라이브러리를 사용하기 위해서는 터미널 창에서 pip install 명령어로 해당 라이브러리를 설치해야함
- ✓ 이때, 기존 라이브러리들과의 호환성 및 충돌 여부를 확인하지 않아, 새로운 라이브러리 설치가 제대로 안 되는 경우가 발생함
- ✓ 반면에 아나콘다는 데이터 과학 관련 라이브러리들이 이미 설치되어 있어, 위와 같은 작업이 줄어들 뿐만 아니라, conda라는 패키지 관리자가 현재 환경과 충돌 여부를 설치 시에 확인하여, 문제가 적음

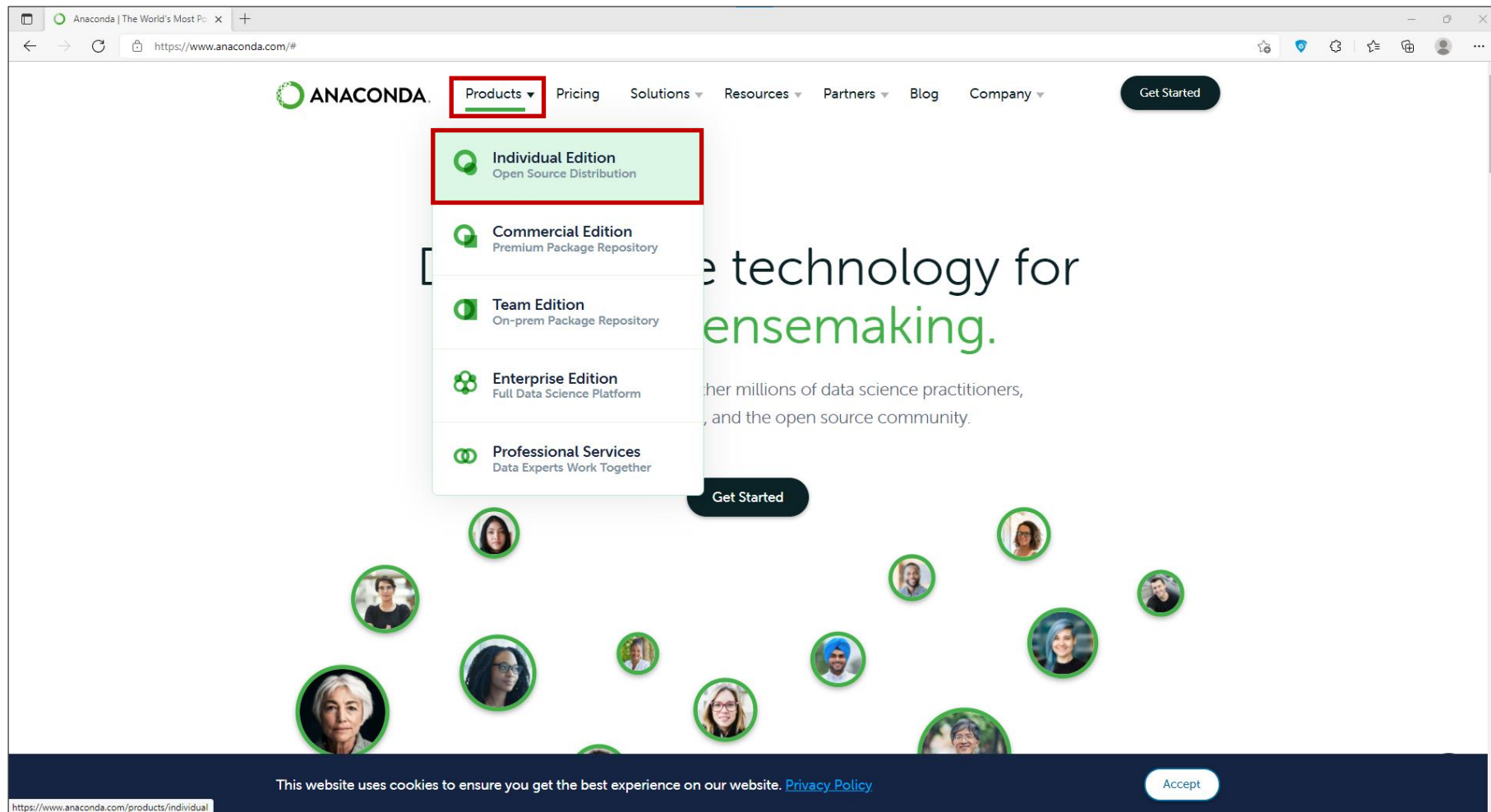
② 수행하는 프로젝트에 따라 가상 환경을 만들어 관리할 수 있음

- ✓ 예를 들어, A 프로젝트에는 python 2를 사용해야 하고, B 프로젝트에는 python 3를 사용해야 하는 경우가 있음
- ✓ 이때, 각 프로젝트에 맞는 가상의 환경을 만들어, 버전에 따른 버그를 방지할 수 있음

# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 설치 (3/12)

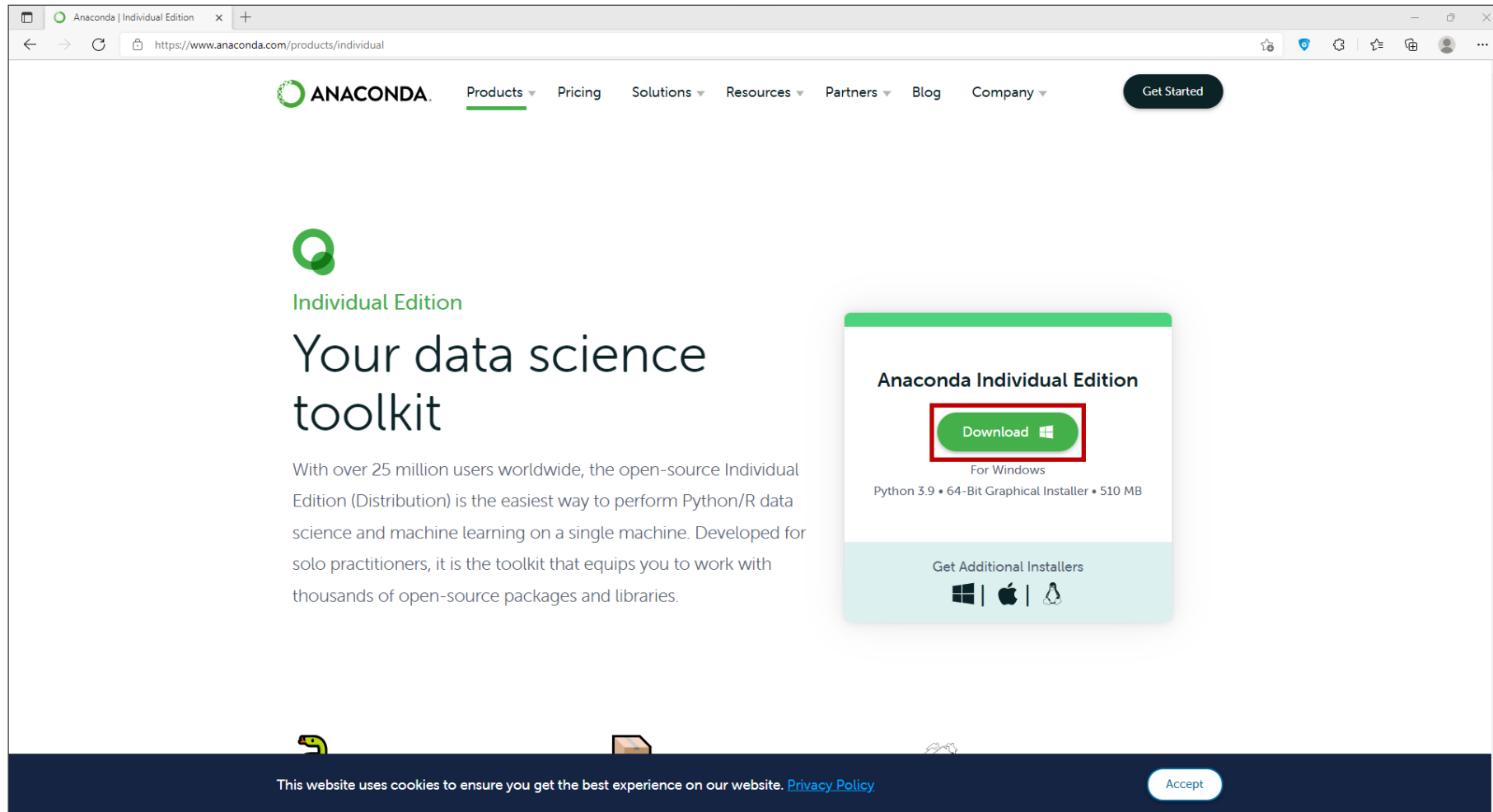
- 아나콘다 사이트(<https://www.anaconda.com/>)에 접속하여, [Products] → [Individual Edition]을 클릭



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 설치 (4/12)

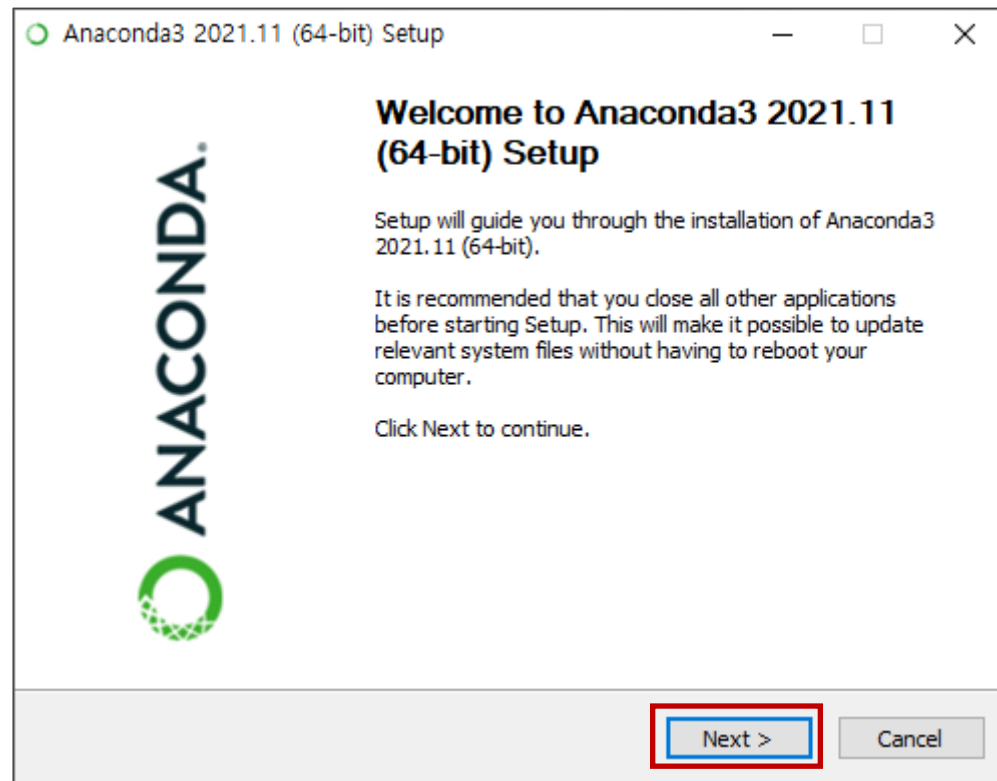
- [Download] 버튼 클릭



## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (5/12)

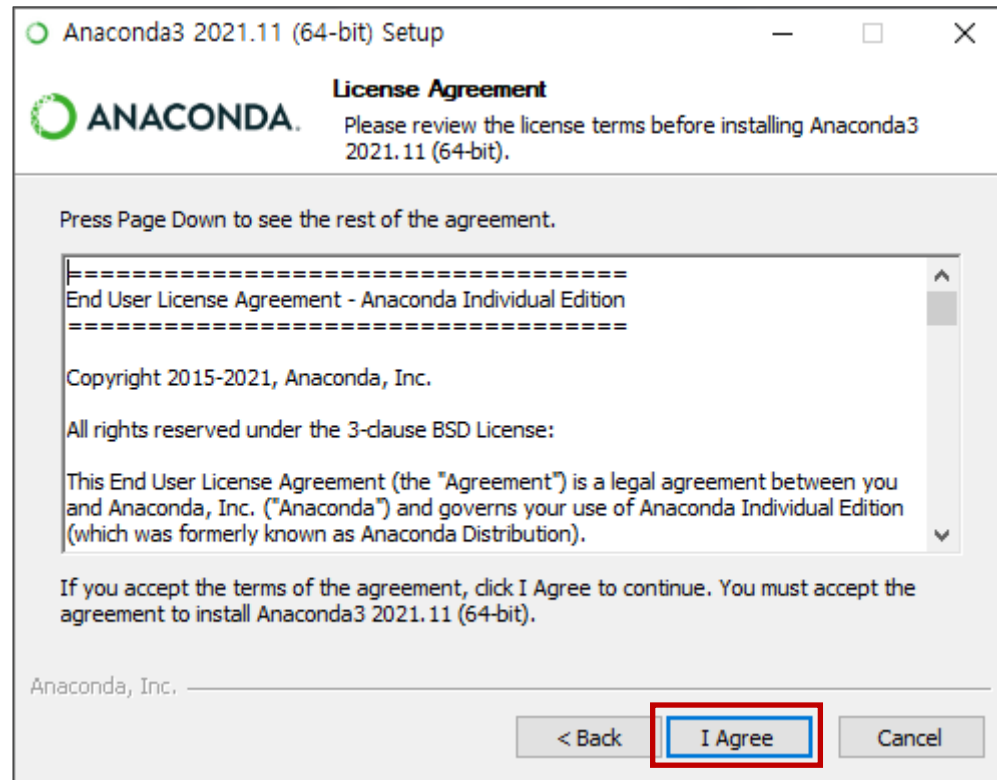
- 내려받은 파일을 실행해 다음 화면이 나오면, [Next] 버튼을 클릭하여 설치를 진행



## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (6/12)

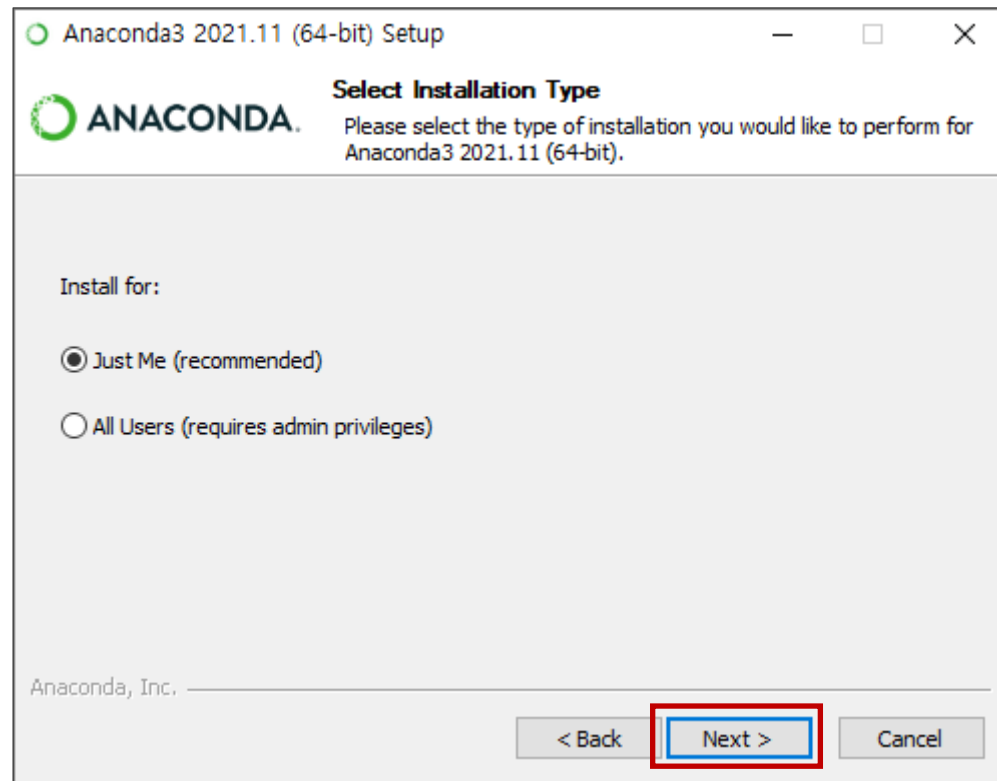
- 라이선스 동의 화면이 나오면, [I Agree] 버튼을 클릭



## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (7/12)

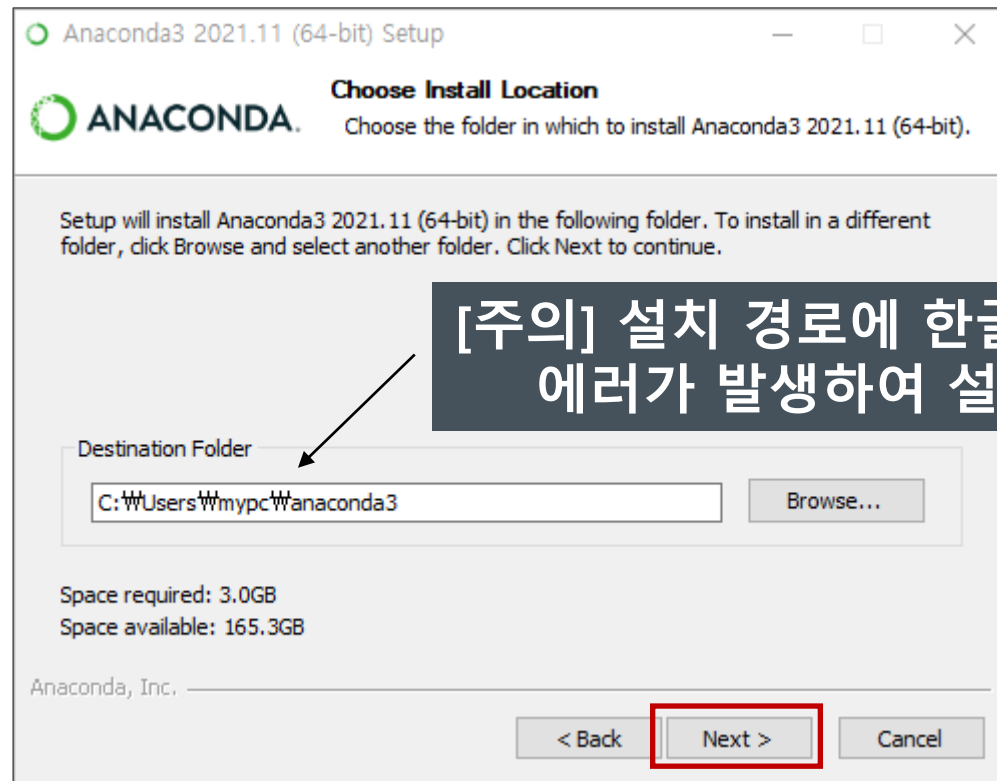
- 설치 타입을 묻는 창이 나오면, [Just Me (recommended)]를 선택하고 [Next] 버튼을 클릭



## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (8/12)

- 설치 경로를 묻는 창이 나오면, 기본값으로 두거나 원하는 경로를 지정하고 [Next] 버튼 클릭

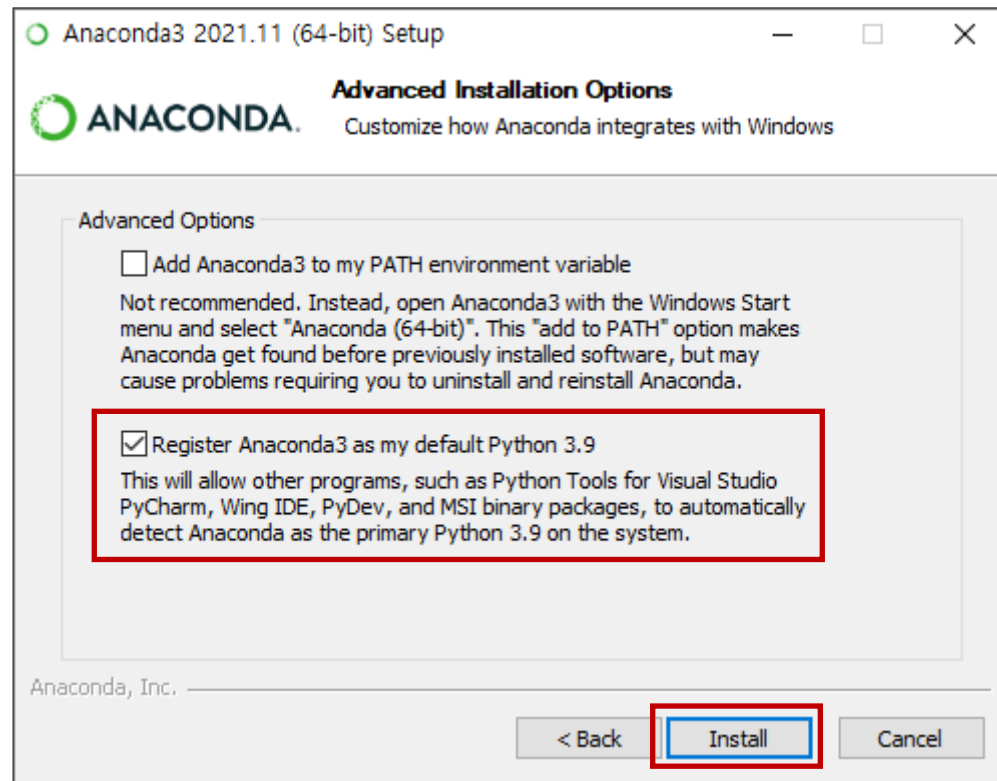


[주의] 설치 경로에 한글이 포함되어 있으면  
에러가 발생하여 설치되지 않습니다!

## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (9/12)

- 옵션을 설정하는 화면이 나오면, 기본값으로 두고 [Install] 버튼을 클릭

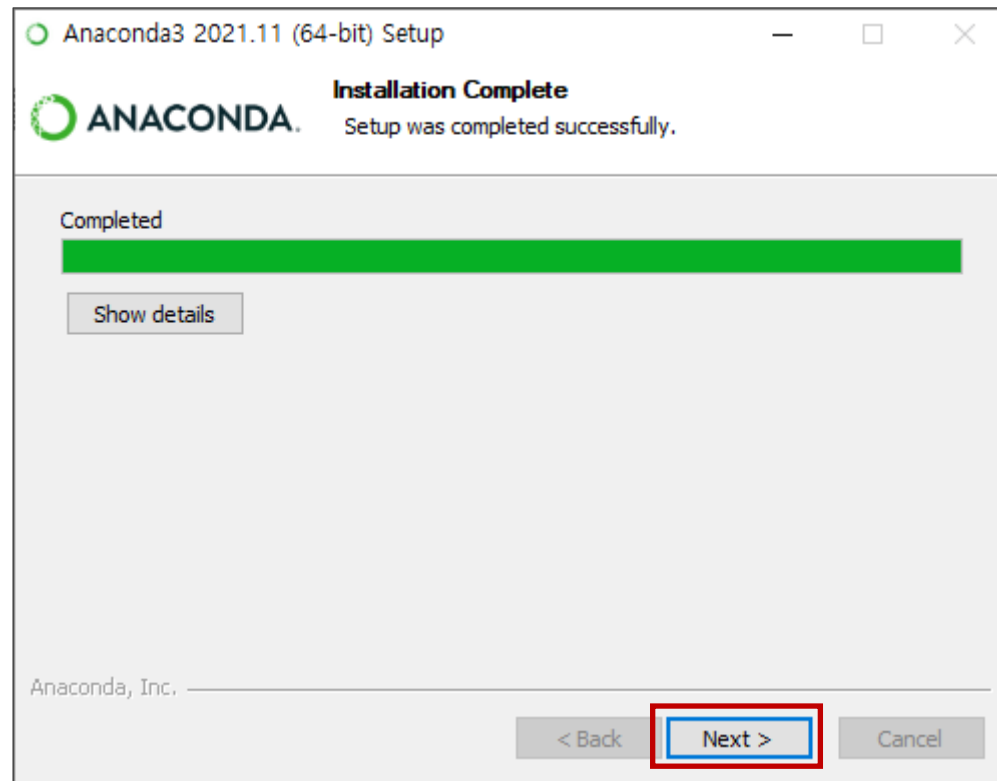




## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (10/12)

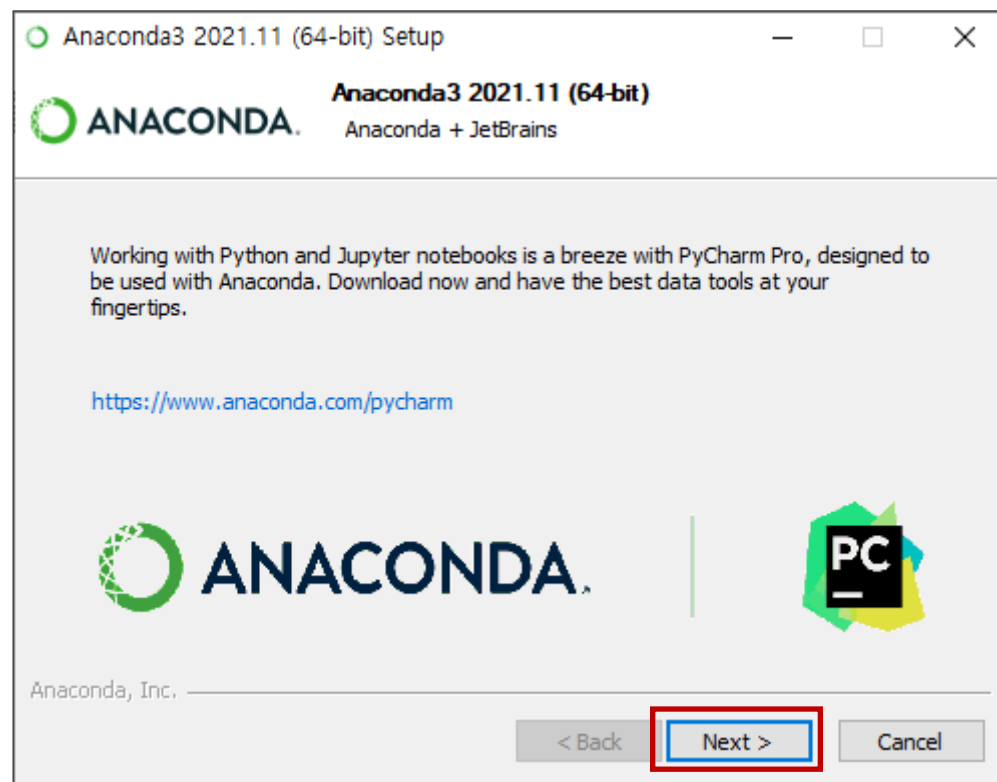
- 설치에는 몇 분 정도 소요되며, 다음과 같이 설치가 진행되면 [Next] 버튼을 클릭



## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (11/12)

- [Next] 버튼을 클릭



## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 설치 (12/12)

- 설치를 마쳤으면 [Finish] 버튼을 클릭



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 아나콘다(Anaconda)

- 과학 연구 및 머신러닝 분야에 적합한 파이썬 및 R 언어의 패키지/의존성 관리 및 배포를 편리하게 해주는 조건부 무료 오픈소스 SW

## ❖ 주피터 노트북(Jupyter Notebook)과 주피터 랩(Jupyter Lab)

- 파이썬 프로그래밍 언어로 코드를 작성하고 실행하는 개발 환경 제공
- 일부 코드의 실행 결과를 확인할 수 있고 문서화하여 다른 사람과 공유 가능
- 아나콘다를 설치하면 주피터 노트북과 주피터 랩을 바로 사용할 수 있음

## ❖ 웹 브라우저

- Microsoft Edge
- Chrome



[사진출처-왼쪽] [https://en.wikipedia.org/wiki/Microsoft\\_Edge#/media/File:Microsoft\\_Edge\\_logo\\_\(2019\).svg](https://en.wikipedia.org/wiki/Microsoft_Edge#/media/File:Microsoft_Edge_logo_(2019).svg)

[사진출처-오른쪽] [https://en.Wikipedia.org/wiki/Google\\_Chrome#/media/File:Google\\_Chrome\\_icon\\_\(February\\_2022\).svg](https://en.Wikipedia.org/wiki/Google_Chrome#/media/File:Google_Chrome_icon_(February_2022).svg)

## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 기본 사용법 (1/14)

- 설치를 완료하여 파이썬 프로그램을 코딩할 준비가 되었으니,  
우선 간단히 Hello, World!를 출력하는 프로그램을 만들어 보자
- 아나콘다를 설치하면 프로그램을 개발할 수 있는 환경이 함께 설치됨
- 이것을 IDE(Integrated Development Environment, 통합 개발 환경)라고 함

# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (2/14)

- 파이썬 프로그래밍을 위한 IDE에는 여러 가지가 있는데, 그 중 널리 사용되는 IDE는 아래와 같음

### ① Visual Studio Code

- Microsoft 사에서 개발한 IDE로 전세계적으로 가장 인기 있는 개발 환경 중에 하나임

### ② PyCharm

- JetBrains 사에서 개발한 IDE로, 코드 컴플리션 기능이 우수하고 파이썬의 장점인 각종 프레임워크 지원이 가장 뛰어남
- Community Edition은 무료
- Professional Edition은 유료

### ③ Spyder

- 오픈 소스 IDE로, PyCharm Professional Edition 같은 IDE를 사용하기 힘든 학생이나 개인 개발자가 사용

아나콘다를 설치할 때 Spyder가 함께 설치됨



[사진출처-왼쪽] [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code#/media/File:Visual\\_Studio\\_Code\\_1.35\\_icon.svg](https://en.wikipedia.org/wiki/Visual_Studio_Code#/media/File:Visual_Studio_Code_1.35_icon.svg)

[사진출처-중앙] [https://en.wikipedia.org/wiki/PyCharm#/media/File:PyCharm\\_Icon.svg](https://en.wikipedia.org/wiki/PyCharm#/media/File:PyCharm_Icon.svg)

[사진출처-오른쪽] [https://en.wikipedia.org/wiki/Spyder\\_\(software\)#/media/File:Spyder\\_logo.svg](https://en.wikipedia.org/wiki/Spyder_(software)#/media/File:Spyder_logo.svg)

## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 기본 사용법 (3/14)

- IDE 대신에 노트북 편집기를 이용해서 파이썬 프로그래밍을 할 수 있음
- 셀(Cell)을 만들어서 셀 안에 코드를 작성해 놓고, 실행시키면서 셀 별로 수행 결과를 확인을 할 수 있음
- 코딩 → 결과 확인 → 코드 수정 → 결과 확인 등의 반복적인 작업에 유용

① Ipython(Interactive Python)

② Jupyter Notebook

③ JupyterLab

④ Google Colaboratory

아나콘다를 설치할 때 Jupyter Notebook과 JupyterLab이 함께 설치됨



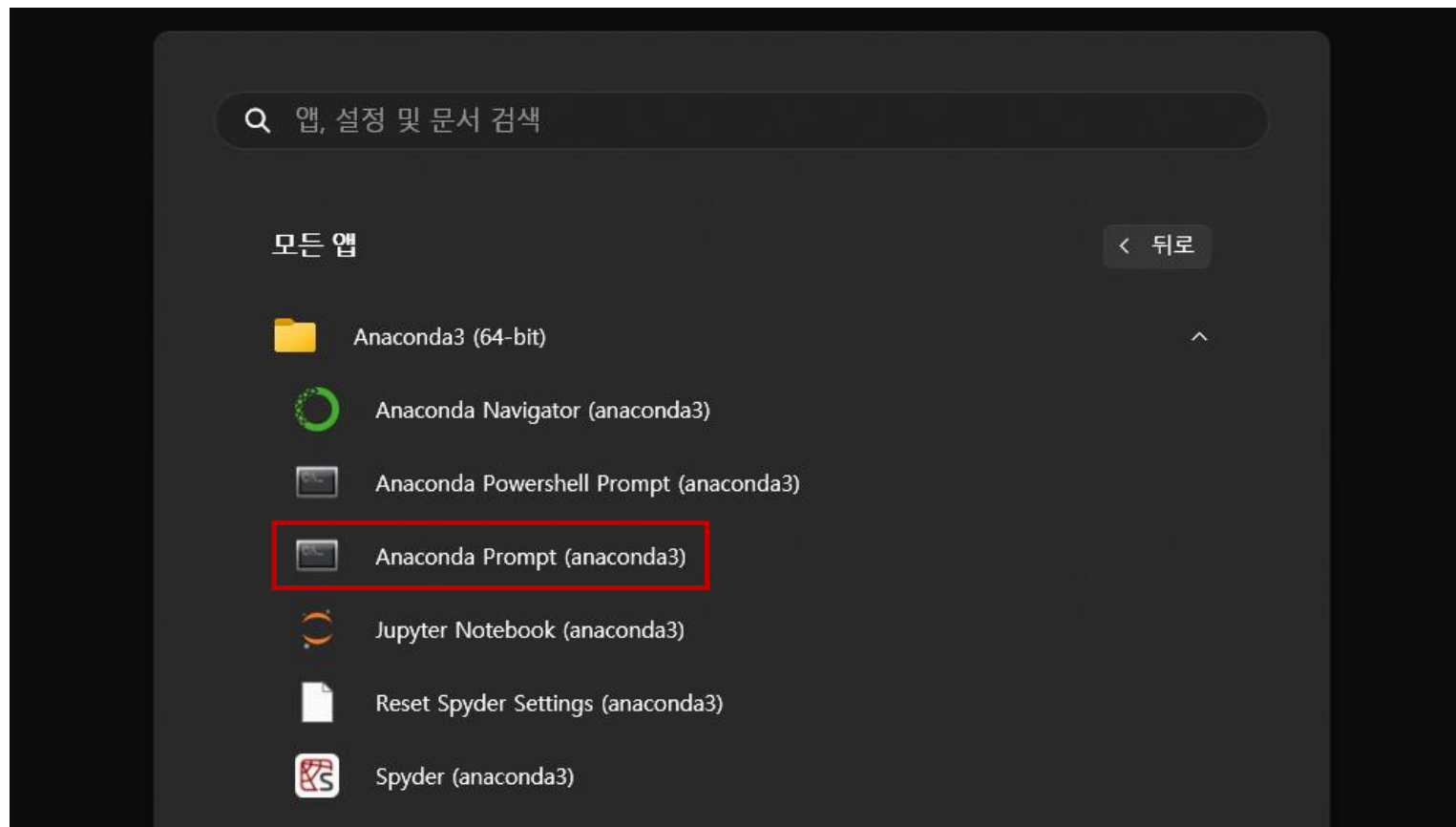
[사진출처-왼쪽] [https://en.wikipedia.org/wiki/Project\\_Jupyter#/media/File:Jupyter\\_logo.svg](https://en.wikipedia.org/wiki/Project_Jupyter#/media/File:Jupyter_logo.svg)

[사진출처-오른쪽] <https://colab.google/>

## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 기본 사용법 (4/14)

- 파이썬 개발 환경인 JupyterLab에서 프로그램을 작성하고 실행하는 연습을 해보자
- 먼저 [시작] – [모든 앱] – [Anaconda3 (64-bit)] – [Anaconda Prompt (anaconda3)]를 클릭
- 또는, [검색]에서 'Anaconda Prompt'를 직접 타이핑하여 아이콘을 찾고, 이를 클릭하여 실행해도 됨

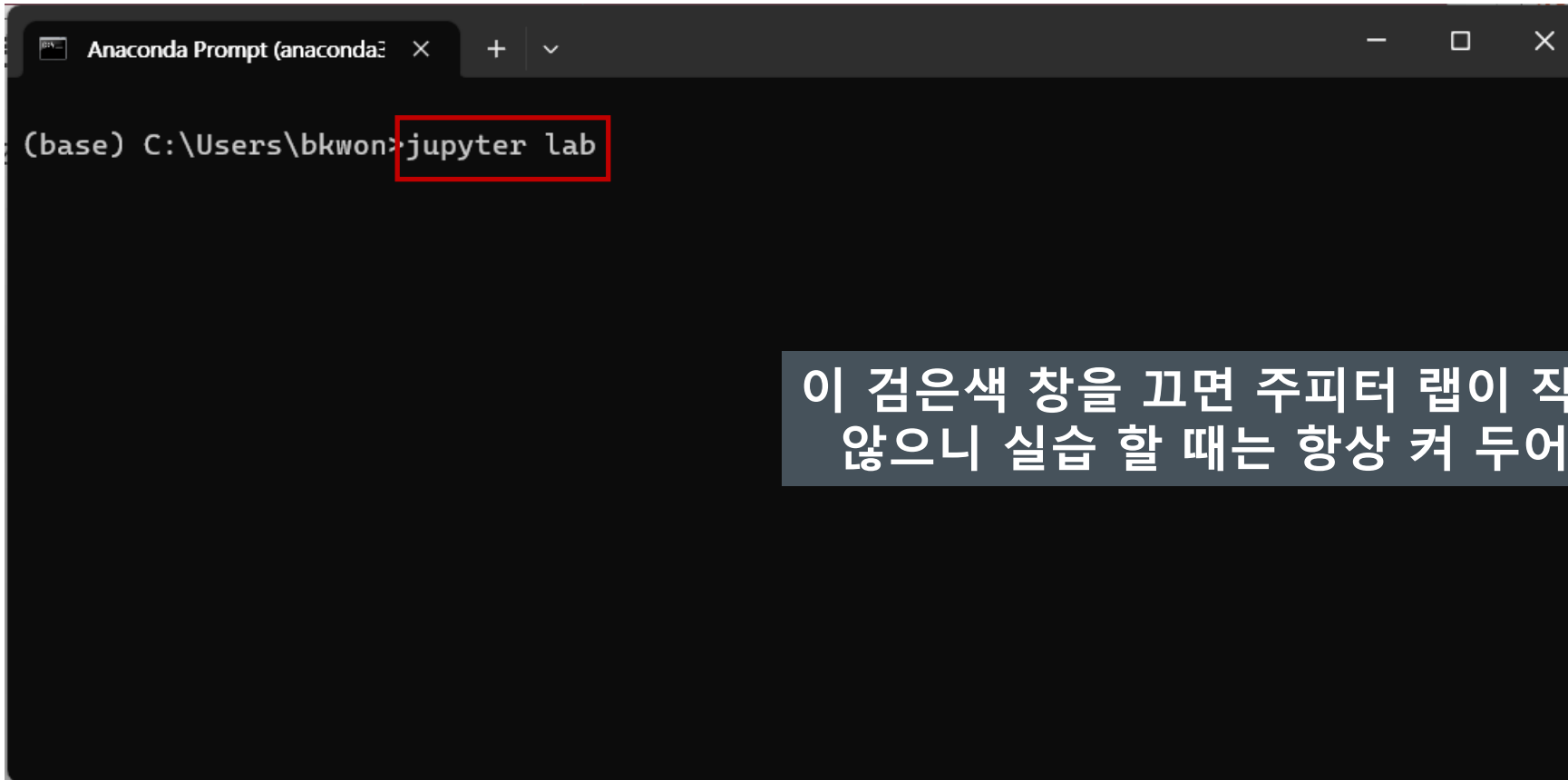




## 03. Standalone 방식의 개발 환경 구축하기

### ❖ 파이썬 기본 사용법 (5/14)

- Anaconda Prompt가 시작되면 아래와 같이 'jupyter lab'을 직접 타이핑한 다음 엔터(Enter) 버튼

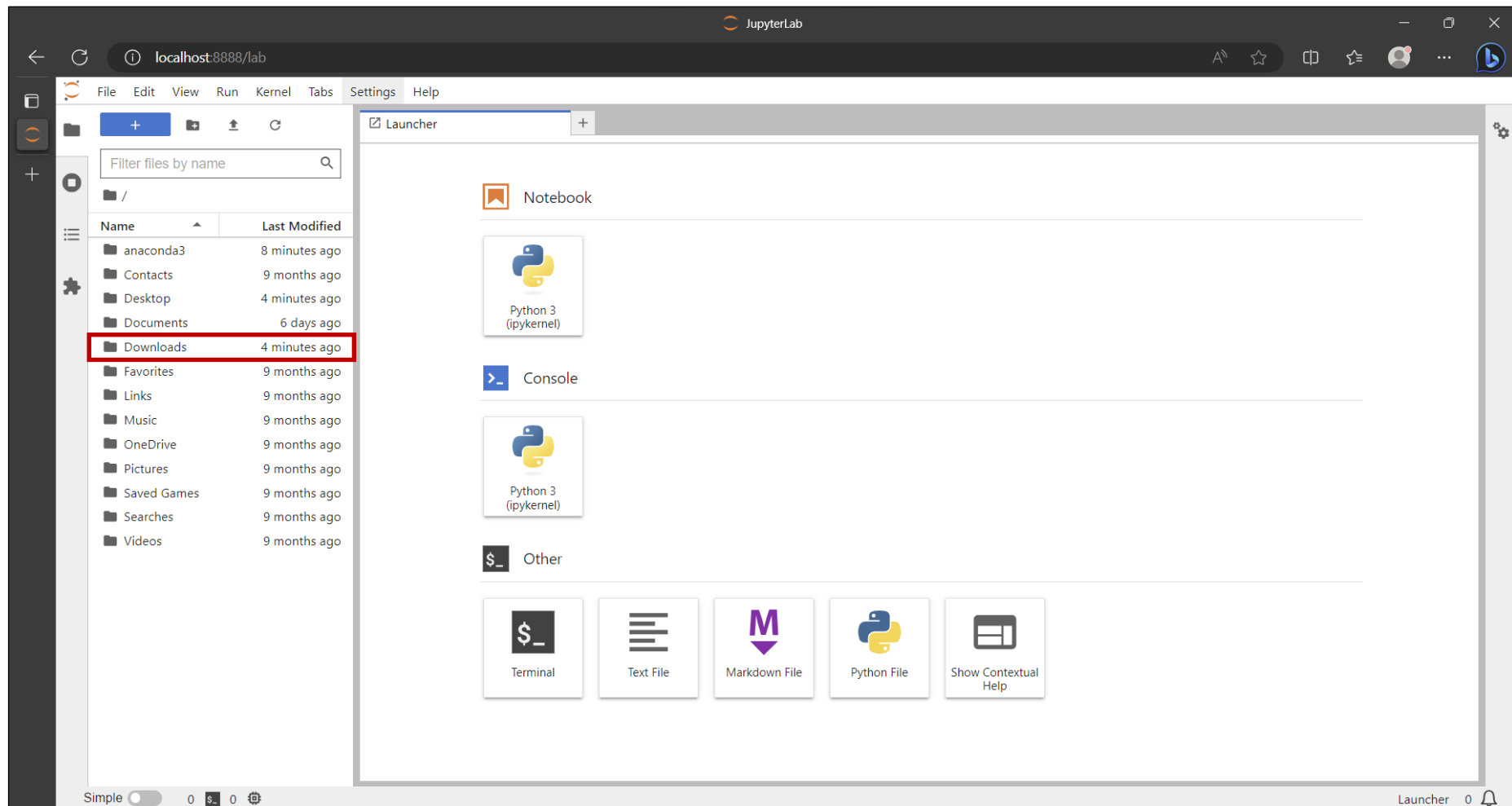
A screenshot of the Anaconda Prompt terminal window. The window title is 'Anaconda Prompt (anaconda3)'. The prompt shows the command '(base) C:\Users\bkwon>jupyter lab' where 'jupyter lab' is highlighted with a red rectangular box. The terminal background is dark, and the text is light gray.

이 검은색 창을 끄면 주피터 랩이 작동되지  
않으니 실습 할 때는 항상 켜 두어야 함

# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (6/14)

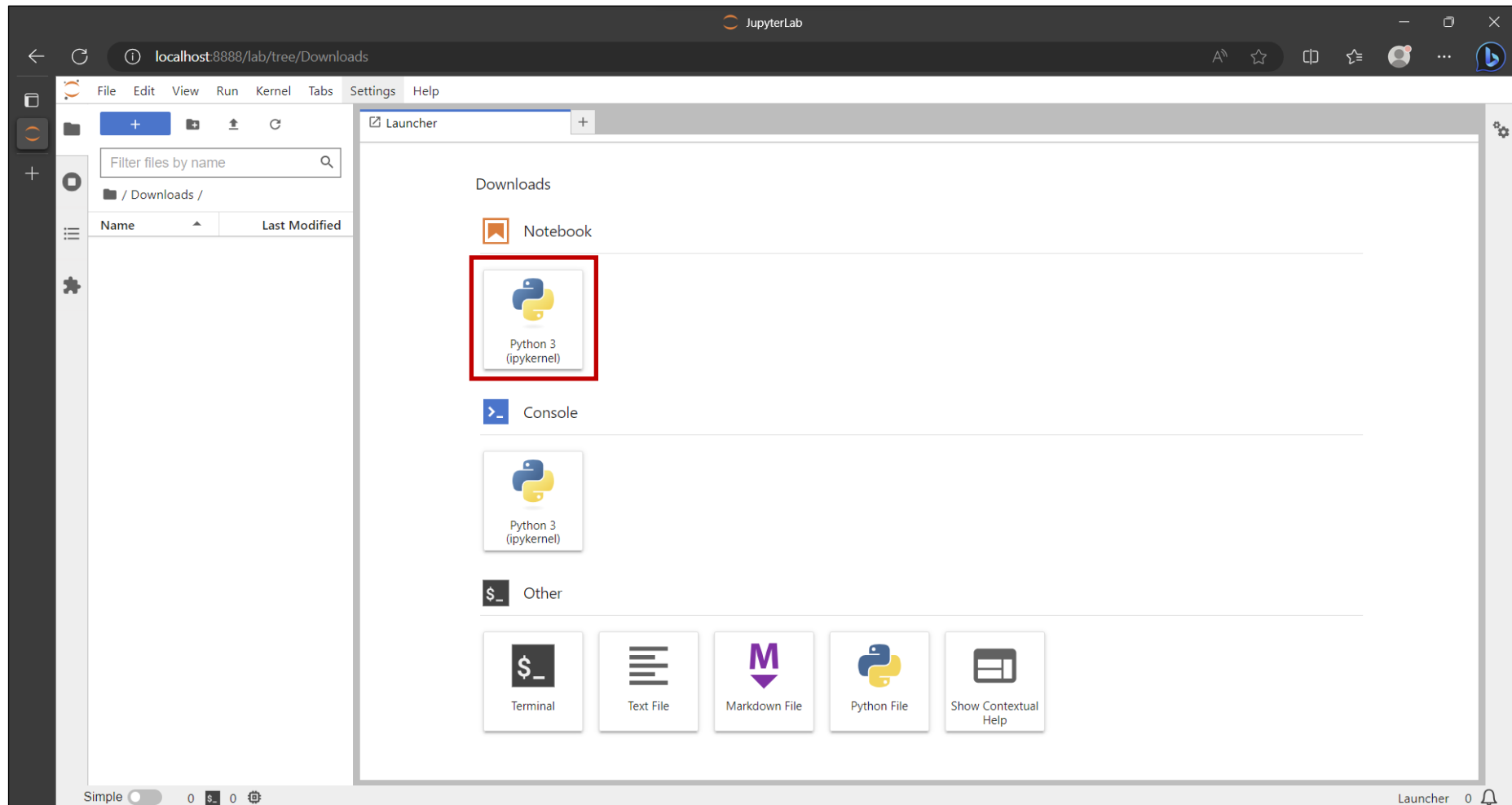
- 웹 브라우저가 켜지면 아래와 같은 화면이 나오고, [Downloads] 폴더를 더블클릭



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (7/14)

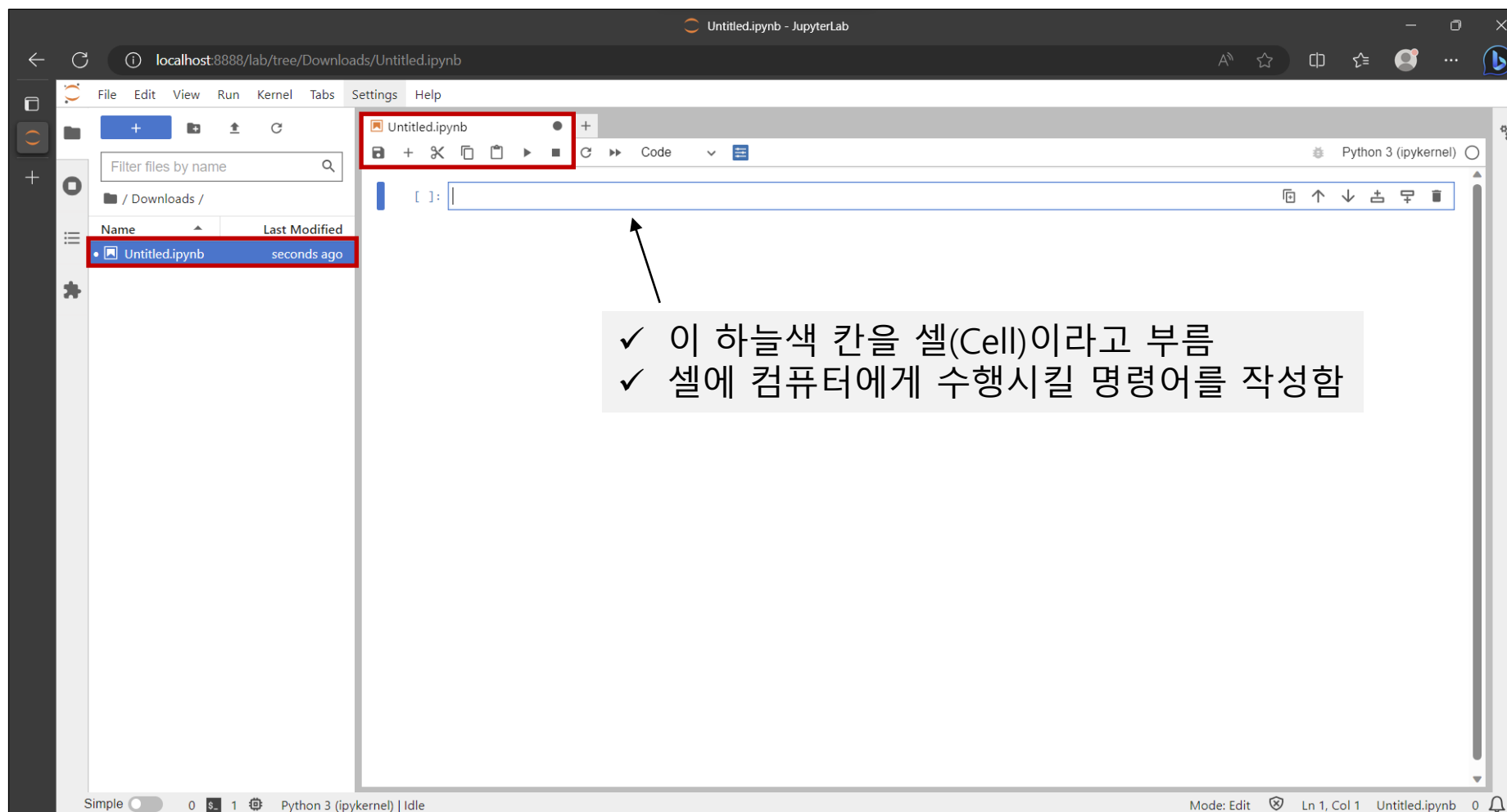
- [Notebook] 바로 아래의 [Python 3 (ipykernel)]을 클릭



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (8/14)

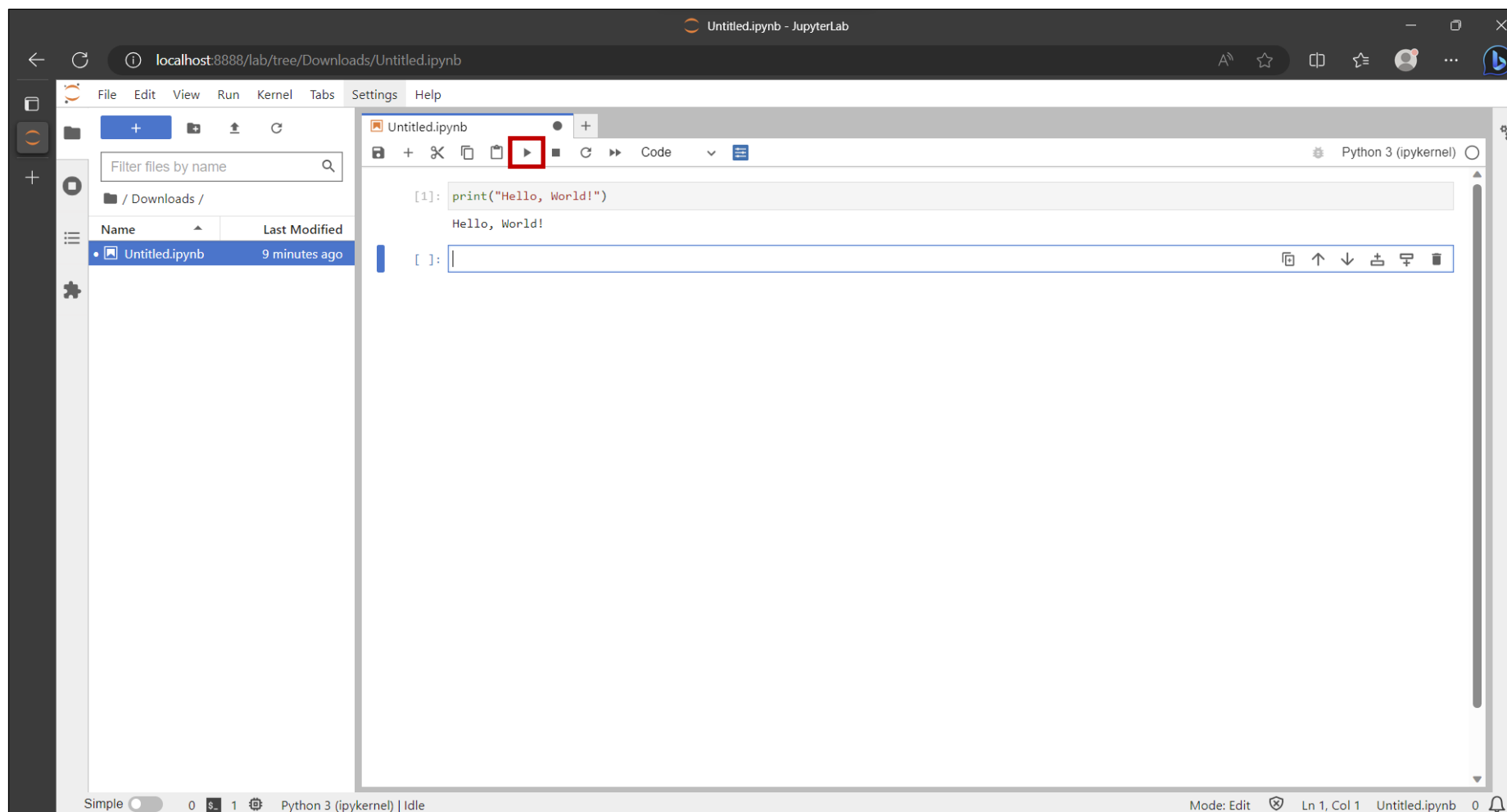
- Untitled.ipynb 파일이 생성된 것을 확인



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (9/14)

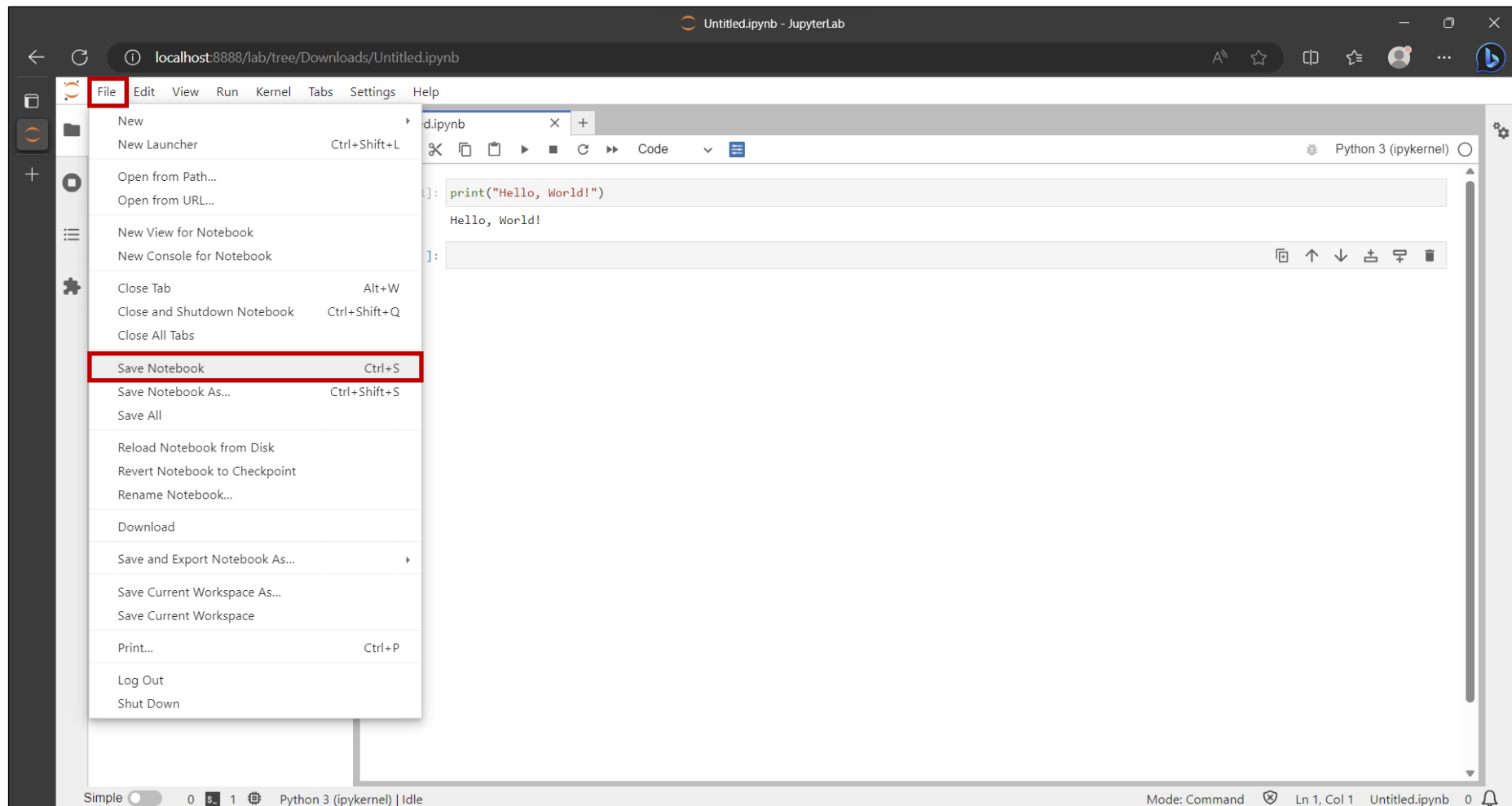
- 셀(Cell)에 `print("Hello, World!")`를 입력하고, ▶ 버튼을 클릭하여 셀을 실행



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (10/14)

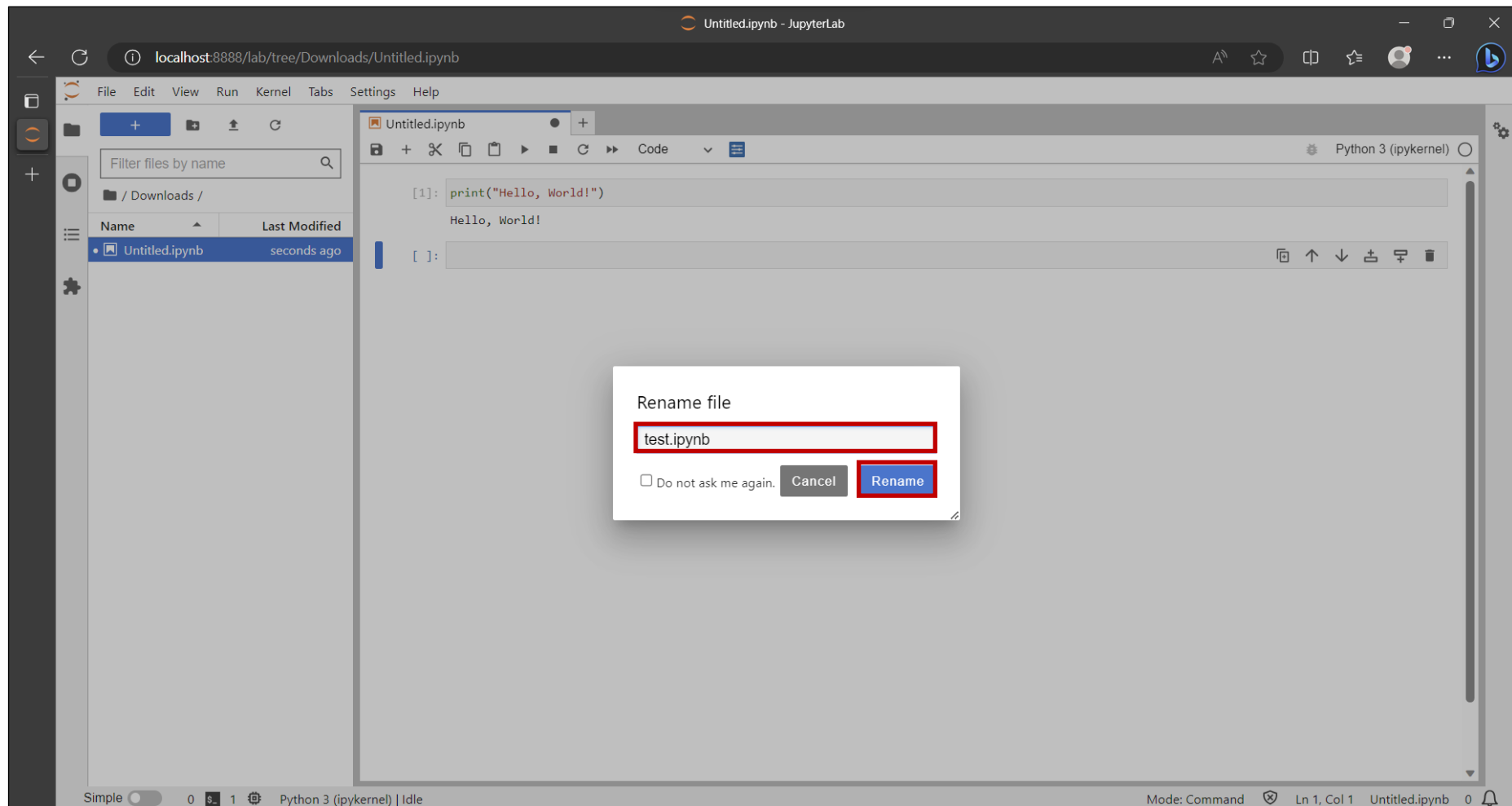
- 작성한 .ipynb 파일을 저장하기 위해서, [File] – [Save Notebook] 버튼 클릭



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (11/14)

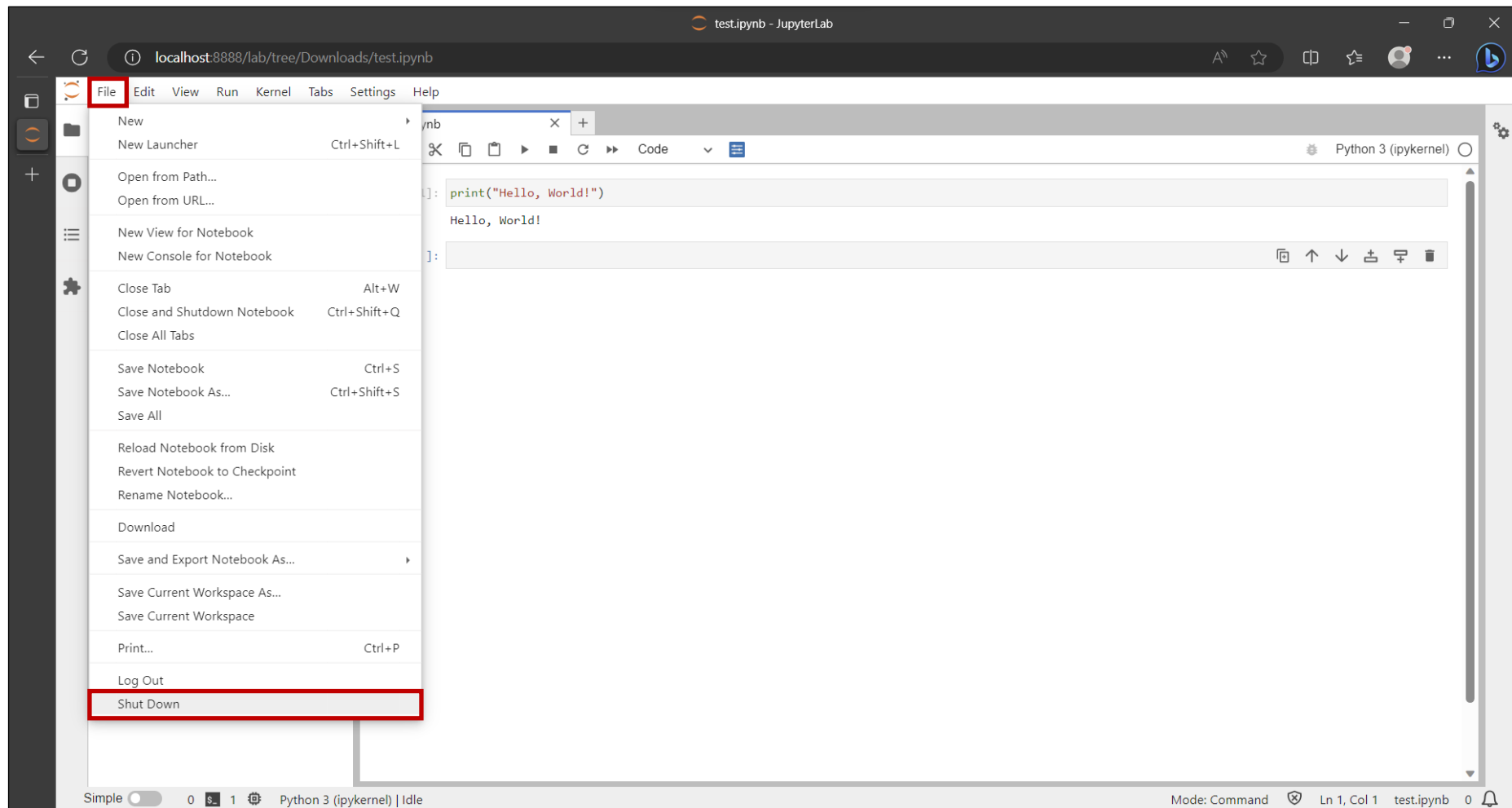
- Untitled를 지우고, 원하는 파일명(여기서는 test)을 입력한 다음 [Rename] 버튼 클릭



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (12/14)

- JupyterLab을 종료하기 위해서, [File] – [Shut Down] 버튼 클릭

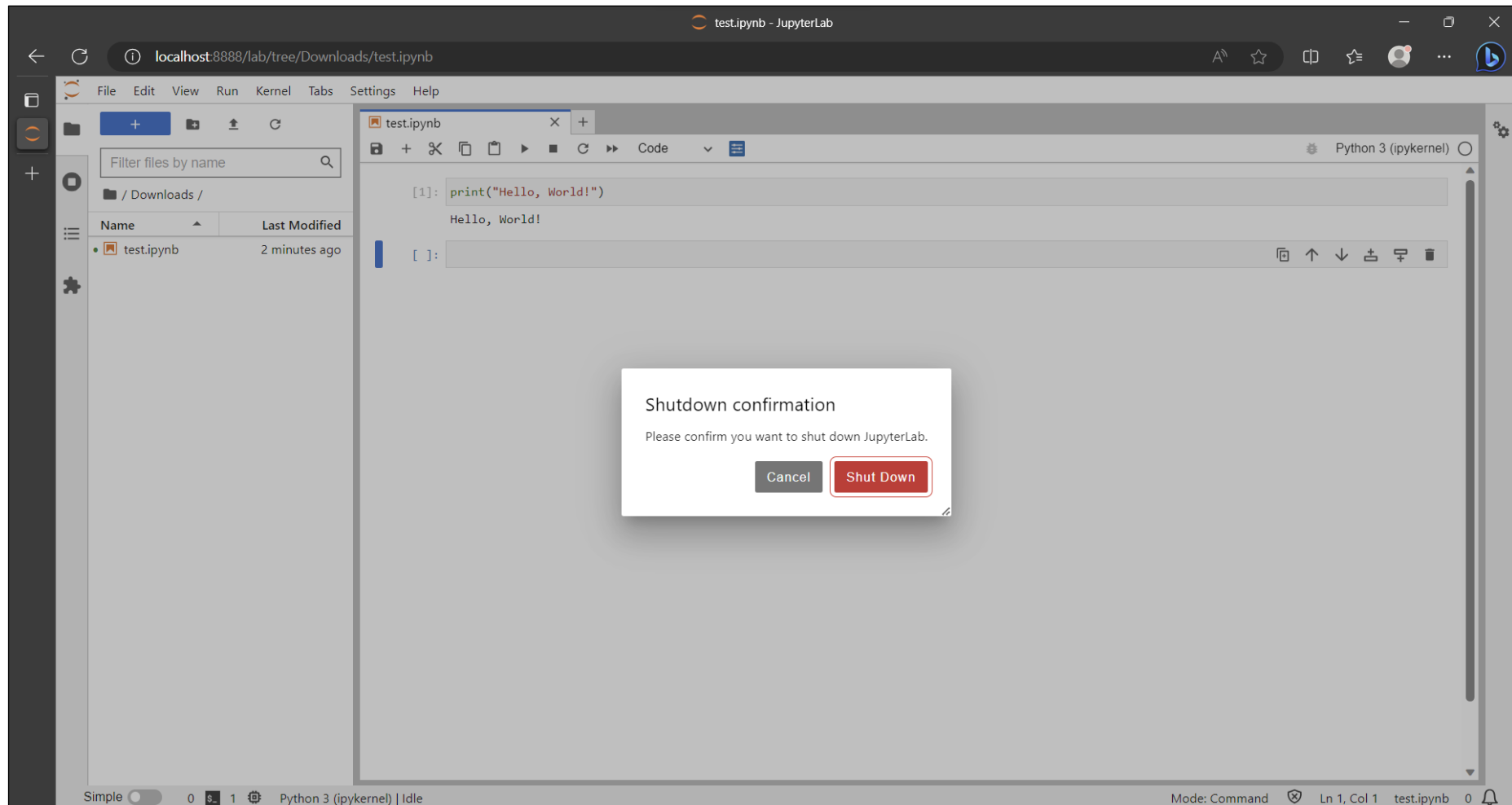




# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (13/14)

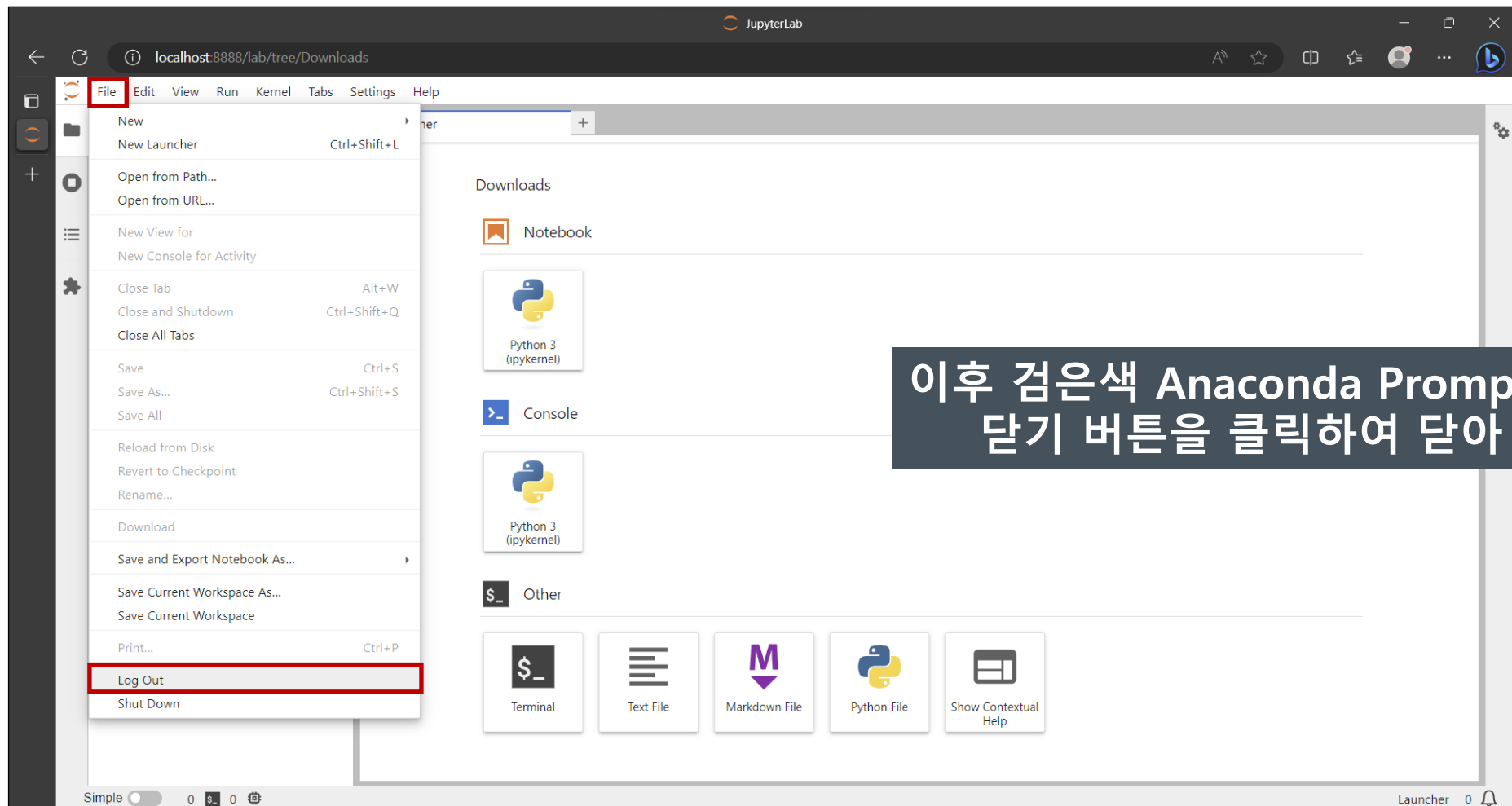
- [Shutdown confirmation] 창이 뜨면, [Shut Down] 버튼 클릭



# 03. Standalone 방식의 개발 환경 구축하기

## ❖ 파이썬 기본 사용법 (14/14)

- [File] – [Log Out] 버튼 클릭하고, 종료되면 해당 웹 탭을 닫아서 종료



- ❖ 01. 두 가지 프로그래밍 환경
- ❖ 02. Cloud 방식의 개발 환경 구축하기
- ❖ 03. Standalone 방식의 개발 환경 구축하기

# THANK YOU!

## Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: [bkwon@dongduk.ac.kr](mailto:bkwon@dongduk.ac.kr)