



프로그래밍 개요

Python Programming

국민대학교 경영대학원 AI빅데이터전공
문현실 (hsmoon@kookmin.ac.kr)





Ch.01

프로그래밍과 Python

프로그래밍의 정의

- 프로그램(Program)
 - 컴퓨터가 어떤 일을 하도록 절차나 방법을 정해둔 것
 - 컴퓨터가 이해할 수 있는 언어로 지시
- 프로그래밍 언어
 - 이러한 절차나 방법을 컴퓨터가 (사람도) 알아들을 수 있도록 만든 인공 언어
- 프로그래밍(Programming)
 - 프로그래밍 언어를 사용하여 프로그램을 만드는 것 ≡ 코딩(Coding)

프로그래밍적 사고

- 프로그래밍 언어의 학습은
 - 하나의 도구를 익히는 과정 +
컴퓨터가 절차적으로 문제를 어떻게 해결하는지 논리적 과정을 학습
 - “이 나라의 모든 사람이 프로그램을 배워야 한다. 왜냐하면 프로그램이라는 것은
생각을 어떻게 하는지 가르쳐 주기 때문이다.” - 스티브잡스
- Programmatic Thinking
 - Computational Thinking
 - Algorithmic Thinking
 - 절차적으로 문제를 해결하는 논리적 과정

프로그래밍적 사고

- 프로그래밍적 사고 절차
 - 분해 : 복잡한 문제를 작은 문제들로 나눔
 - 패턴인식 : 기존 해결책을 재사용하는 것이 가능한지 확인
 - 추상화 : 문제를 단순화하여 표현
 - 알고리즘 : 가장 효율적인 절차를 설계

프로그래밍적 사고

- 프로그래밍적 사고의 효과
 - 컴퓨터를 내 마음대로 제어
 - 생각하는 방법의 변화
 - 창의적인 사람
 - 논리적인 사고방식과 문제 해결력 향상

프로그래밍 언어

- 프로그래밍 언어의 종류
 - 기계가 이해하기 쉬울수록 저수준(low-level)
 - 사람이 이해하기 쉬울수록 고수준 (High-level)

언어	
기계어	가장 저수준, CPU에서 사용하는 명령
어셈블리어	기계어와 1:1로 대응되는 프로그래밍 언어
C	일반적으로 사용하는 저수준 언어. 시스템 프로그래밍 등에 사용
C++	C와 호환되면서 고수준 언어의 측면을 가짐. 게임 개발 등에 사용
Java	C++보다 좀 더 고수준 언어. 기업용 소프트웨어 개발 등에 사용
Python	초보자도 쉽게 배워서 쓸 수 있도록 만들어진 초고수준 언어. 웹 개발, 데이터 분석 등에 사용
R	통계 계산을 위한 초고수준 언어

- 기계어= 어셈블리 \approx C > C++ \approx Java >> Python > R

Python은?

- 1991년 네덜란드 귀도 반 로섬 (Guido Van Rossum)이 개발
 - 1989년 크리스마스에 할 일이 없어 혼자 집에 있다가 개발
 - 2018년 BDFL(Benevolent Dictator For Life) 사임
 - 코미디 프로그램 'Monty Python's Flying Circus'
 - 처음에는 C언어 기반으로 개발되어 이후 다양한 기능이 추가적으로 개발



Python의 대표 철학

- 아름다운 게 추한 것보다 낫다.
 - Beautiful is better than ugly
- 명시적인 것이 암시적인 것 보다 낫다.
 - Explicit is better than implicit
- 단순함이 복잡함보다 낫다.
 - Simple is better than complex
- 복잡함이 난해한 것보다 낫다.
 - Complex is better than complicated
- 가독성은 중요하다.
 - Readability counts

Python의 대표 철학

- 프로그래밍은 프로그래머가 자신의 아이디어를 표현하고 소통하는 방법
 - 이 때의 청중은 컴퓨터가 아니라 다른 프로그래머



Python의 특징

- 간결하고 알기 쉬운 문법과 강력한 기능
- 오픈소스
 - 방대한 사용자와 활발한 커뮤니티
- 크로스 플랫폼 (Windows, Mac OS X, Linux)
- 대화형 환경
- 스크립트 언어
- 높은 생산성

Python의 특징

- 인터프리터 언어

- 컴파일러(Compiler) vs 인터프리터(Interpreter) ≡ 번역 vs 통역

구분	컴파일러	인터프리터
작동 방식	소스코드를 기계어로 먼저 번역하고, 해당 플랫폼에 최적화되어 프로그램을 실행	별도의 번역 과정 없이 소스코드를 실행 시점에 해석하여 컴퓨터가 처리할 수 있도록 함
장점	실행 속도가 빠름	간단히 작성, 메모리가 적게 필요
단점	한 번에 많은 기억 장소가 필요함	실행 속도가 느림
주요 언어	C, 자바(Java), C++, C# 등	Python, R 등

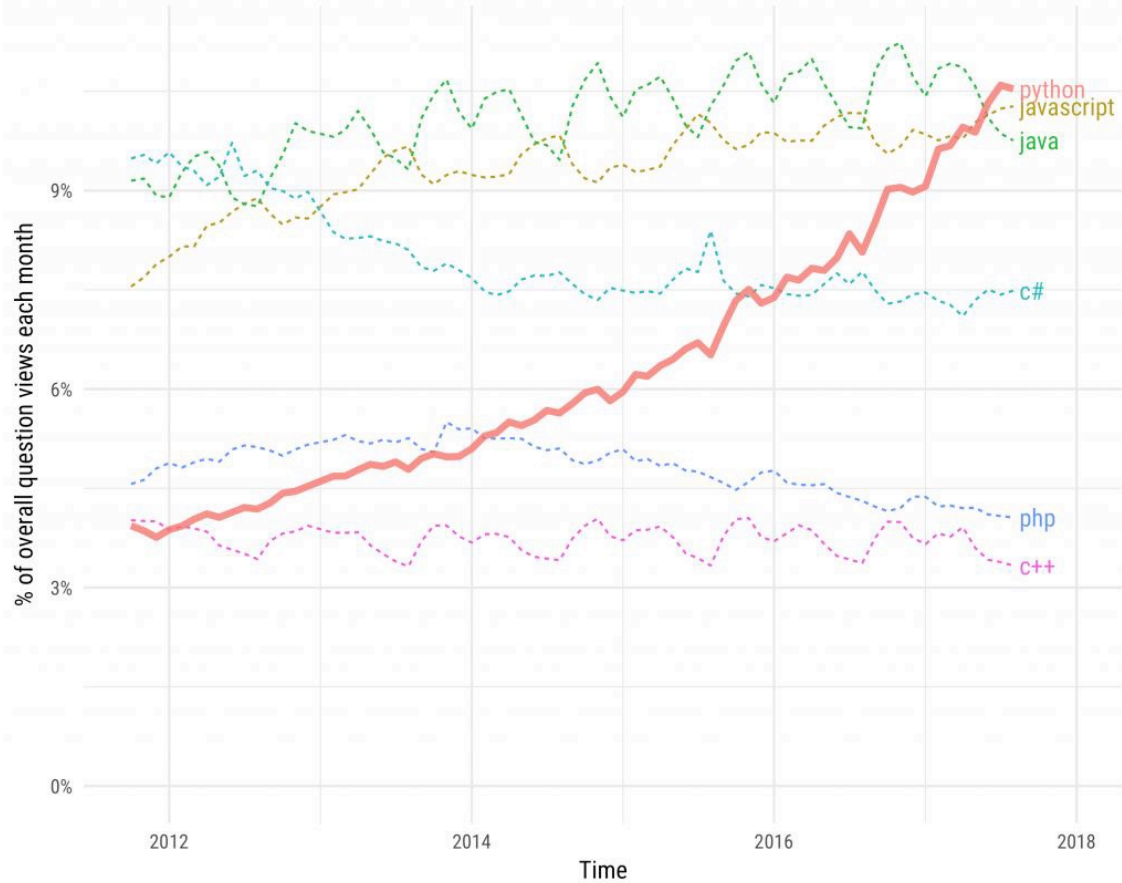
Python의 특징

- 객체 지향 언어
 - vs. 절차적 프로그래밍 언어(Procedure programming)
 - 실행 순서대로 하나부터 열까지 프로그래밍 하는 것
 - 객체 지향 언어는 해당 프로그램이 해결해야 할 문제의 구성요소를 요소별로 정의
 - 각 요소의 기능(method, 메서드)과 정보(attribute, 속성)를 정의
 - 요소들을 결합하는 프로그램을 작성하는 방식
 - 요소 = 객체(object)
 - 이미 만들어진 객체를 재사용할 수 있는 재사용성(reusability)의 특징

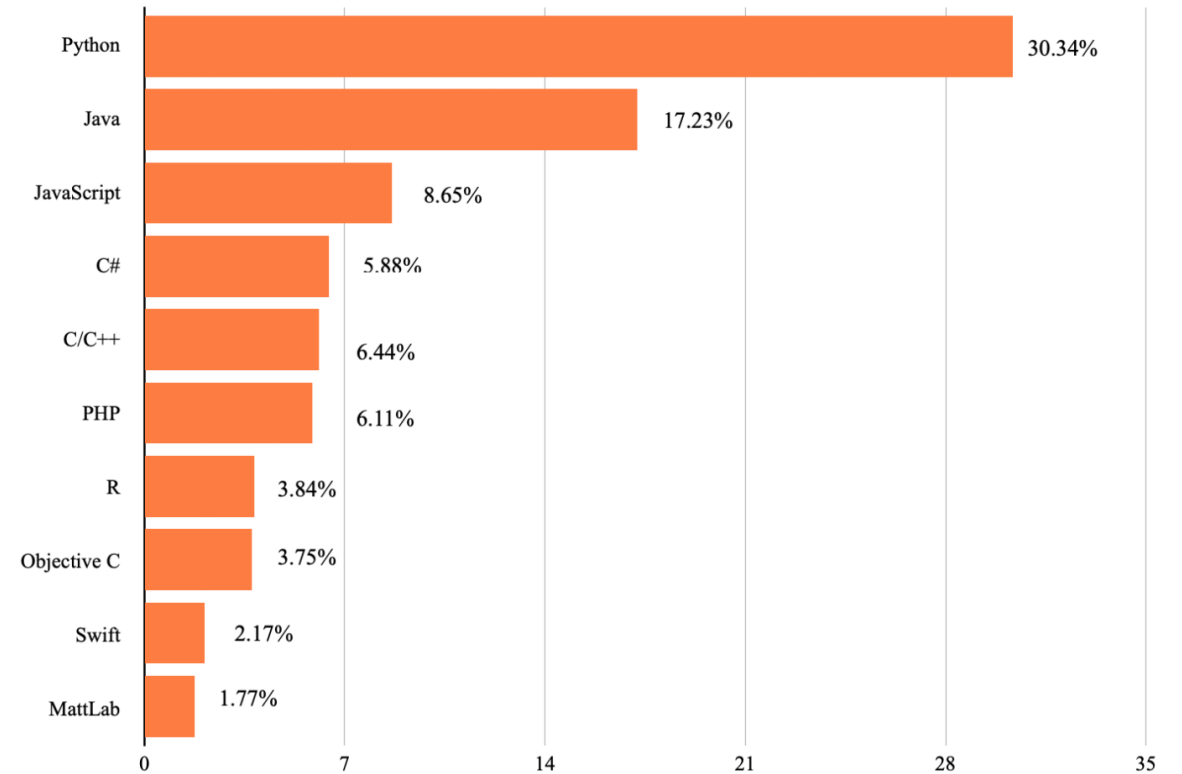
Python 활용 분야

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



The popularity of Programming Language (PYPL) Ranking 2020

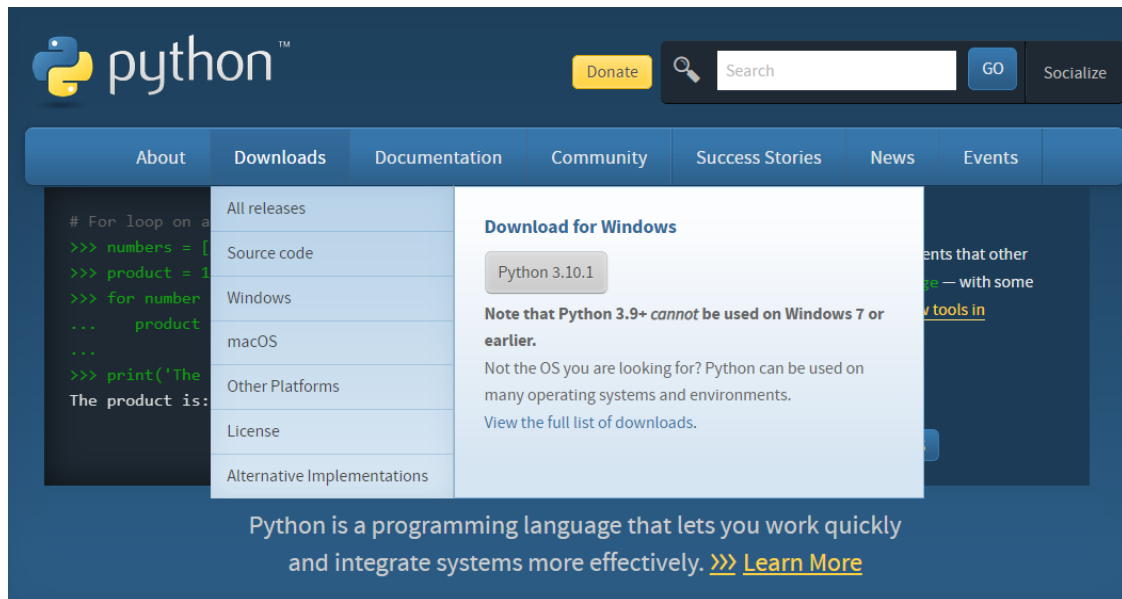


Python 활용 분야

- 웹 개발
- 업무 자동화
- 데이터 분석
- 인공지능/머신러닝
- 과학 계산
- 텍스트 처리
- 이미지 처리
- 데이터베이스
- 구글 , 유튜브 , 인스타그램 , 드롭박스 , NASA 등에서 사용

Python 버전

- 최신버전 : 3.10.1 (2021년 1월 기준)
 - 호환성 문제로 2.x버전도 사용





Ch.02 Python 환경 구축

Python 사용 방법

- Python (<https://www.python.org>)
 - 기본버전
- Anaconda
- Google Colaboratory

Anaconda

- <https://www.anaconda.com>
- 데이터 분석, 기계 학습 등을 위한 Python 배포판
- Python 다양한 패키지 각종 편의 기능들

Anaconda 다운로드 및 설치

- 다운로드

- 구글에서 “anaconda” 검색
- Get Started -> See all Anaconda products -> Individual Edition
 - 운영체제에 맞게 다운로드

- 설치

- 윈도우는 관리자 권한으로 실행 권장
- 영상 참고

Python IDLE

- 통합개발환경(IDLE; Integrated Development Environment)
 - 파이썬 코드를 작성하고 실행할 수 있는 환경
- 대표적인 IDLE
 - Python 기본 IDLE
 - 메모장, Nano Editor
 - 대화형 실행 환경
 - ipython
 - PyCharm
 - Jupyter Notebook

Python에서 계산하기

- 한 줄에 하나의 명령문(statement)
 - 덧셈 $1+1$
 - 뺄셈 $5-3$
 - 곱셈 $6 * 3$
 - 나눗셈 $7/2$
 - 제곱 $5 ** 2$
 - 몫 $8 // 3$
 - 나머지 $19 \% 5$

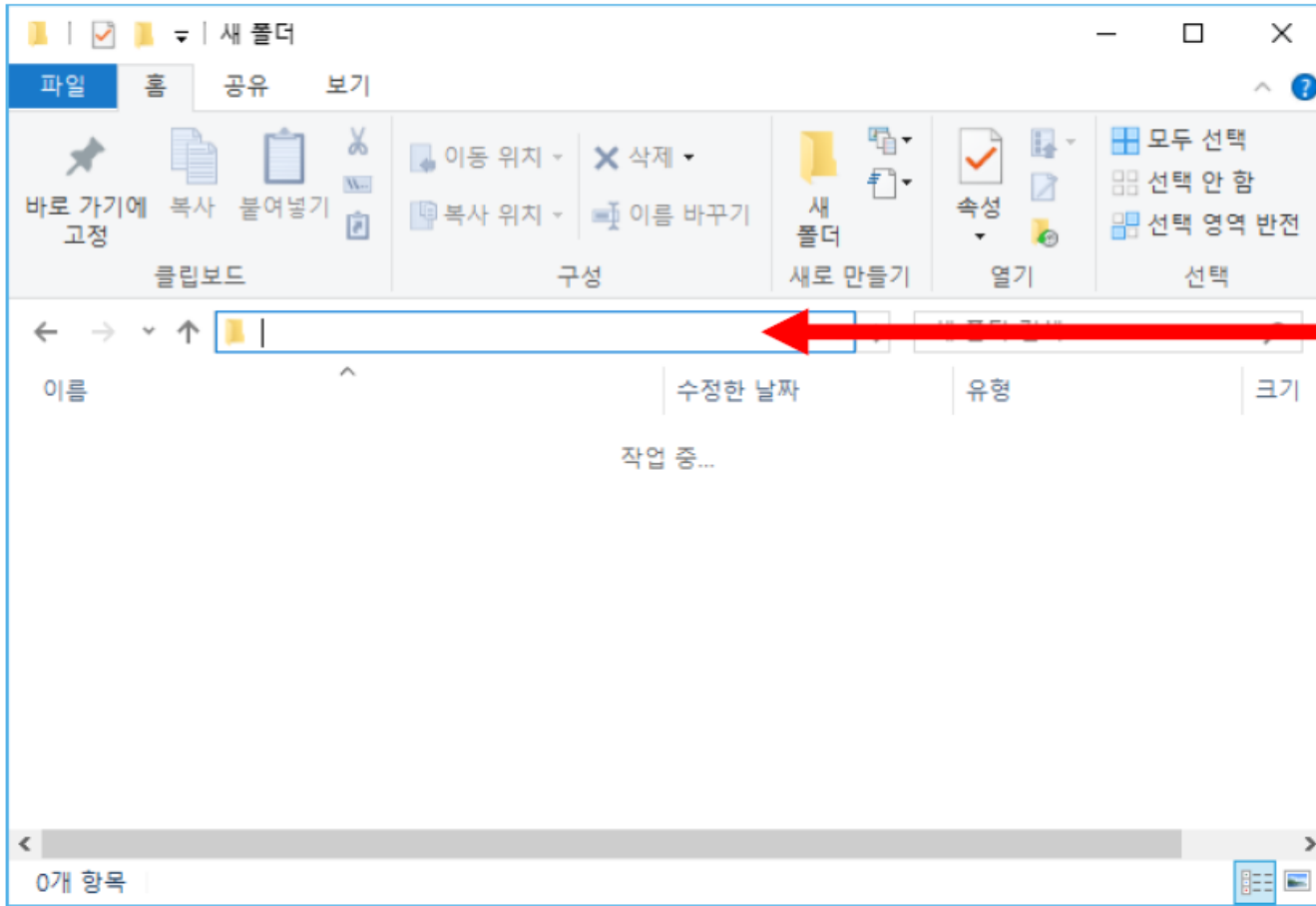


Ch.03 Jupyter Notebook 기초

Jupyter Notebook

- 웹 브라우저를 사용하여 대화형으로 코드 작성 가능
- 코드의 공유가 용이
- 마크업(mark up)으로 문서 형태 작성 가능
- Explorer 미권장

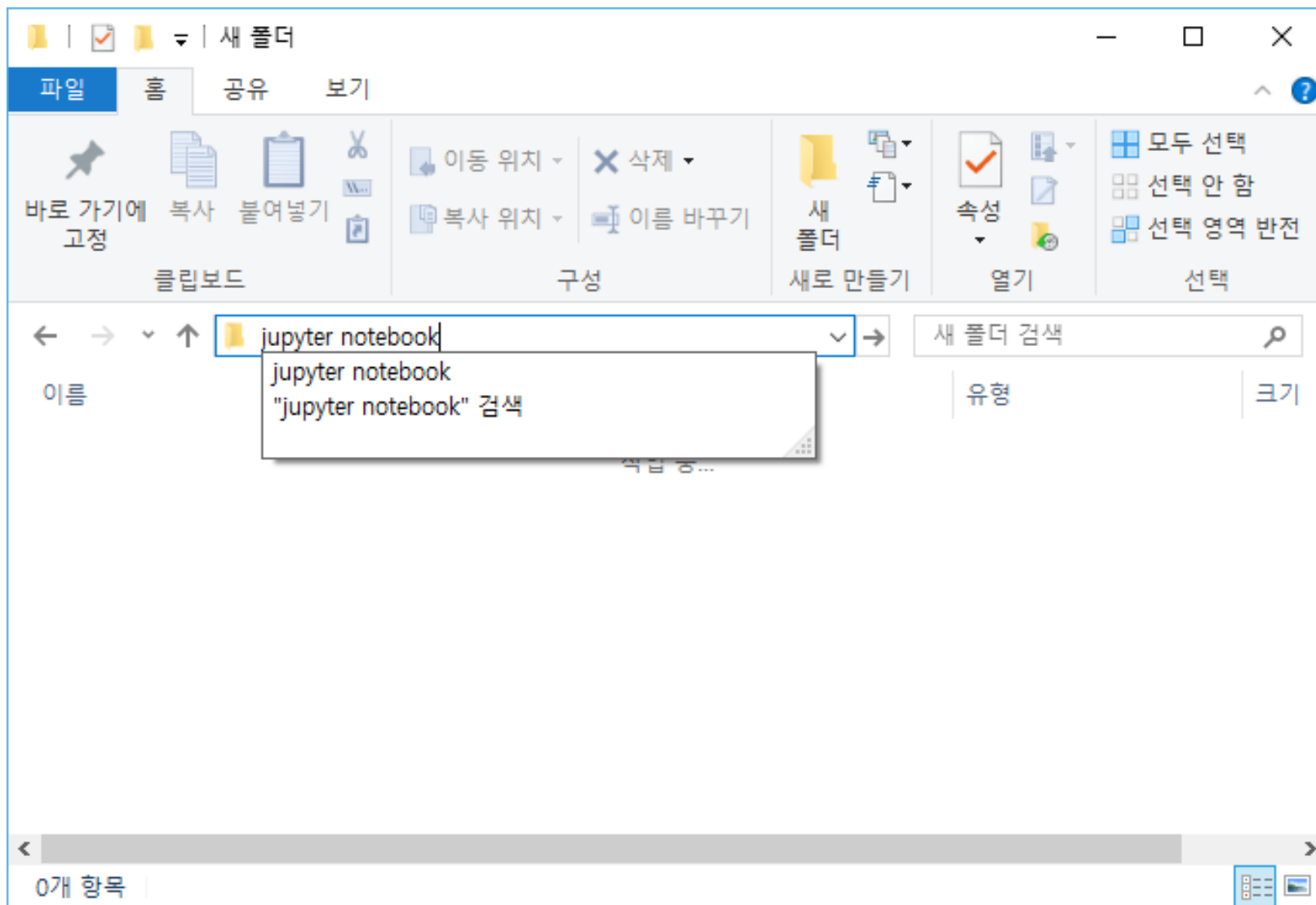
Jupyter Notebook 실행하기(1/4)



탐색기 주소창에
jupyter notebook
입력 후 엔터

- cmd 입력하여 명령창 열거나
- python 또는 ipython 실행도 가능
- 이렇게 실행해야 해당 폴더에서 작업

Jupyter Notebook 실행하기(2/4)



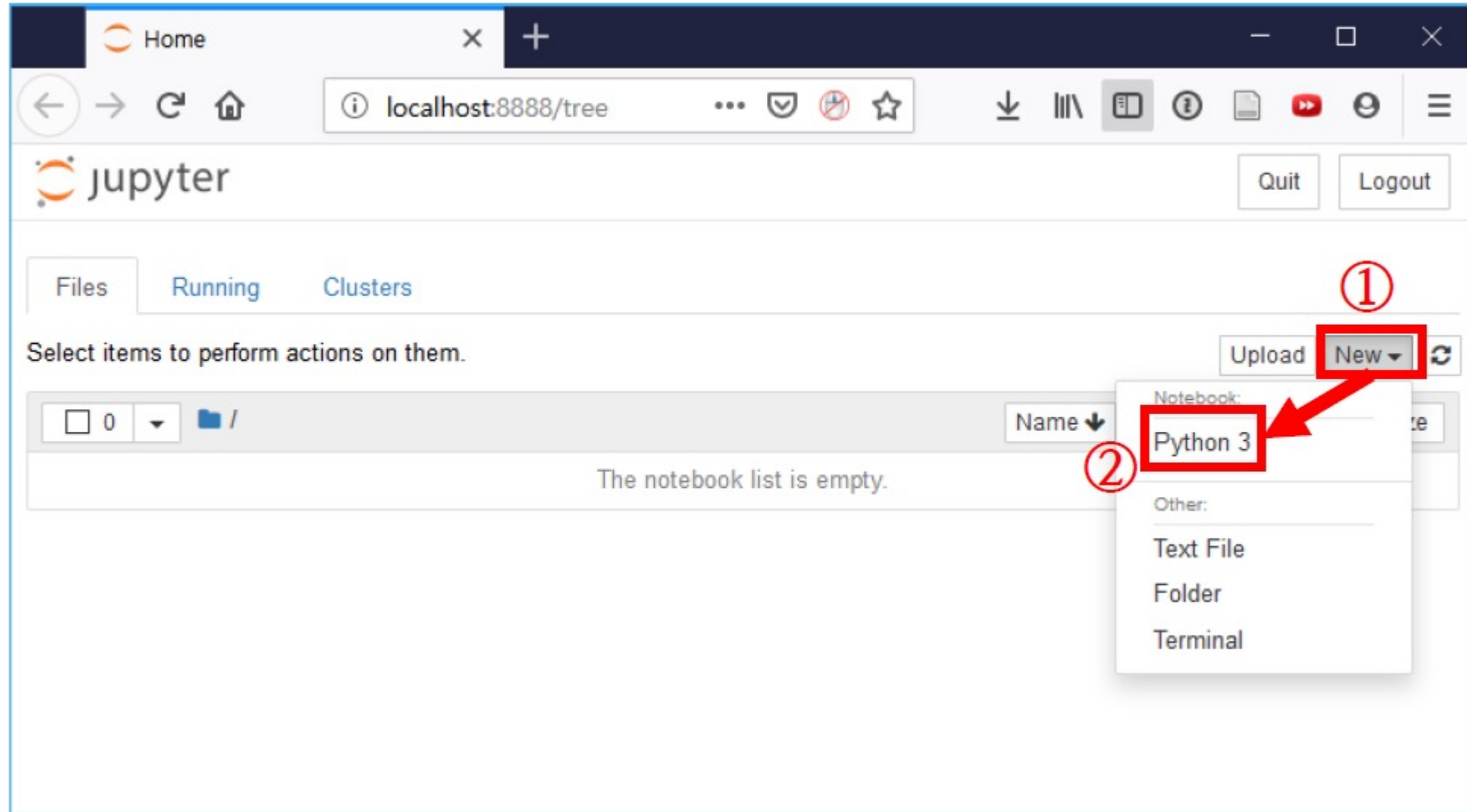
Jupyter Notebook 실행하기(3/4)

```
C:\Users\W eupho\Anaconda3\Scripts\jupyter.exe
[I 18:32:48.507 NotebookApp] [Jupyter Server Extension] Changing NotebookApp.contents_manager_class from LargeFileManager to jupyter.TextFileContentsManager
[I 18:32:48.509 NotebookApp] Serving notebooks from local directory:
[I 18:32:48.510 NotebookApp] The Jupyter Notebook is running at:
[I 18:32:48.510 NotebookApp] http://localhost:8888/?token=fc1ed93b8cd81ea580e451434aa193e9ea52655c5f731b34
[I 18:32:48.510 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 18:32:48.652 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/eupho/AppData/Roaming/jupyter/runtime/nbserver-25500-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=fc1ed93b8cd81ea580e451434aa193e9ea52655c5f731b34
```

jupyter notebook 접속 주소

Jupyter Notebook 실행하기(4/4)

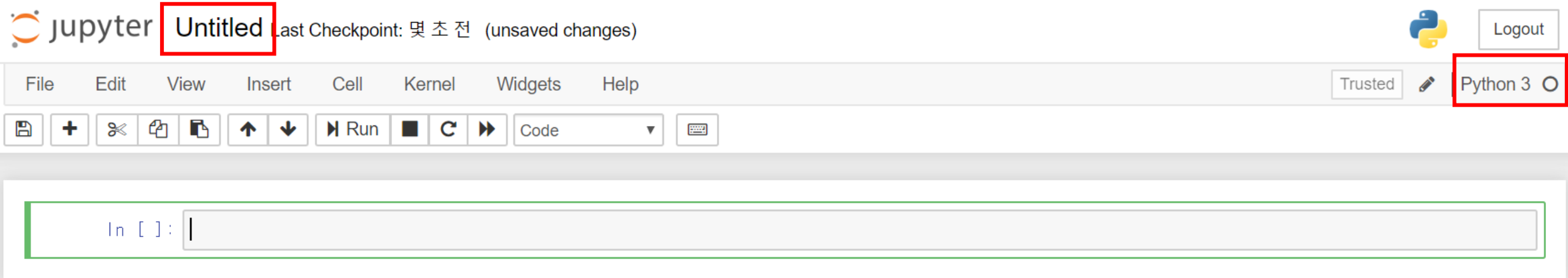


Jupyter Notebook 기초 사용법

- 파일명

- Jupyter Notebook으로 작성한 파일은 .ipynb 확장자
- Google Colab도 동일

- 커널(Kernel) 정상 대기 상태



Jupyter Notebook 기초 사용법

- 셀 모드
 - 셀 자체를 편집

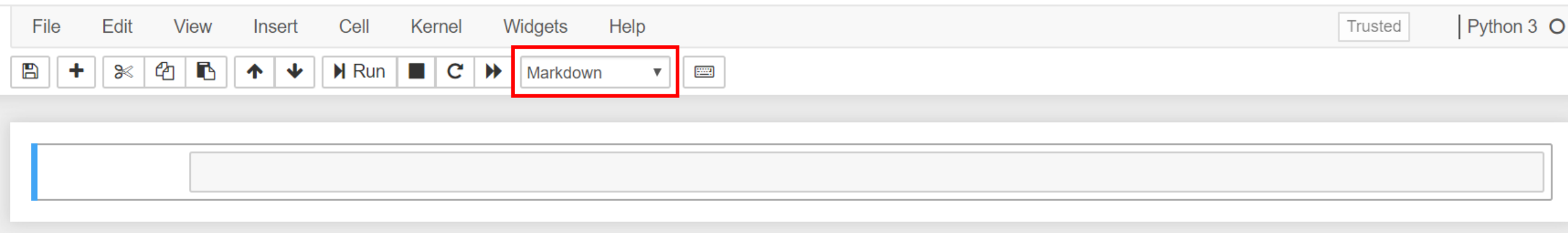
In []:

- 코드 입력 모드
 - 해당 셀에서 Enter 입력 또는 셀 내부 클릭

In []:

Jupyter Notebook 기초 사용법

- 셀 종류
 - 일반 코드 : In [] 이 앞에 표시
 - 주석이 필요할 경우 #
 - 마크다운 : 셀모드에서 M 또는 shortcut에서 Markdown으로 변경
 - 자유롭게 텍스트 입력 가능 : 프로그램 실행에 영향을 미치지 않음



Jupyter Notebook 기초 사용법

- 셀 추가/삭제
 - 셀 추가
 - Insert 메뉴 또는 셀모드에서 A (앞에 셀 추가) 또는 B (뒤에 셀 추가)
 - 셀 삭제
 - Edit -> Delete Cells 또는 셀모드에서 D+D
 - 셀 삭제 취소 : 삭제 뒤 바로 Z

Jupyter Notebook 기초 사용법

- 셀 내용 실행
 - Cells 메뉴 활용
 - Ctrl + Enter : 해당 셀 실행
 - Shift + Enter : 해당 셀 실행 후 아래에 셀 하나 추가

Jupyter Notebook 기초 사용법

- Kernel 메뉴
 - Interrupt : 강제로 실행 중지
 - Restart : 커널 재시작
 - Restart & Clear Output : 출력 결과를 모두 지우고 커널 재시작
 - Restart & Run All : 커널 재시작 후 전체 코드 실행
 - Reconnect : 실행 중인 커널에 접속
 - Shutdown : 커널 강제 종료

마크다운(Markdown) 기본 문법

- 간단한 구조의 문법을 사용하여 웹에서 콘텐츠를 생성
 - 간결하고 별도의 도구 없이 작성 가능하며 다양한 형태로 변환 가능
- Jupyter Notebook에서는 주석 용도
 - 주석은 코드의 해석을 쉽게해주는 설명 : 프로그래머들을 위한 것
 - 특정 코드를 실행하지 않게 하는 용도로도 사용
 - Python 코드의 주석은
 - 한줄일 경우 : #을 활용
 - 여러줄일 경우 : 큰따옴표3개 또는 작은 따옴표 3개

마크다운(Markdown) 기본 문법

- 제목(Headers)
 - #
- 인용문자(BlockQuote)
 - >
- 순서 있는 목록
 - 숫자.
- 순서 없는 목록
 - *, +, -

마크다운(Markdown) 기본 문법

- 수평선
 - ***
 - ---
- 강조
 - *이탤릭* 또는 _이탤릭_
 - **굵게** 또는 __굵게__
- 줄바꿈
 - 문장 마지막에서 3칸 이상 띄어쓰기



Ch.04 Google Colaboratory 사용법

Google Colab

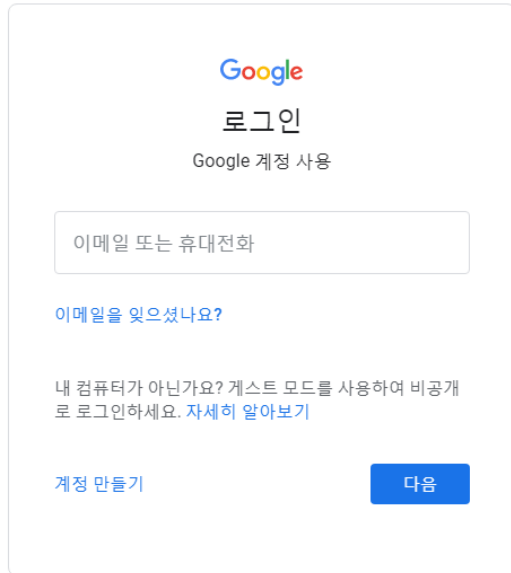
- <https://colab.research.google.com>
- 구글에서 제공하는 무료 Python 사용 환경
- 고성능 기계학습에 필요한 설정 완비
- 구글 드라이브와 연동
- Chrome 권장

Google Colab의 한계

- 시간 제한
 - 하나의 작업을 최대 12시간까지만 연속 사용 가능
 - 1 시간 이상 미사용시 초기화
- 서버에 원격 접속하여 작업
 - 대용량 데이터 분석이 불편
 - 외부 반출할 수 없는 데이터는 분석할 수 없음

Google Colab 사용하기

- 구글 드라이브 접속 (drive.google.com)



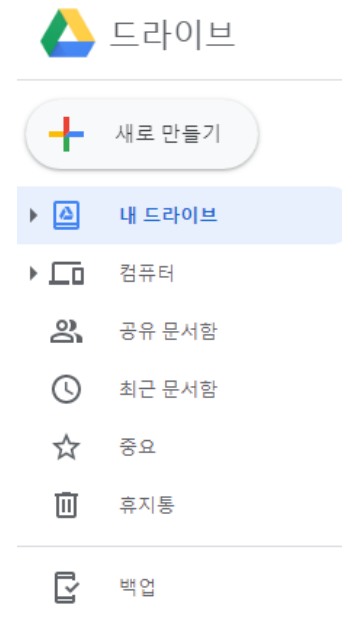
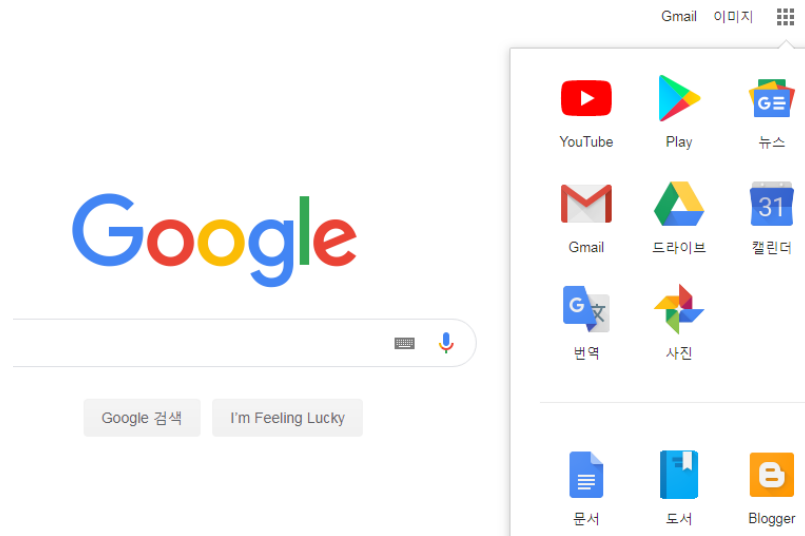
Google
로그인
Google 계정 사용

이메일 또는 휴대전화

이메일을 잊으셨나요?

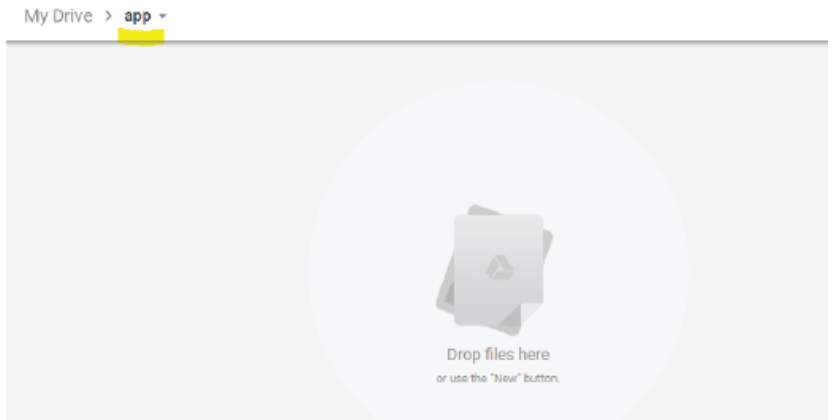
내 컴퓨터가 아닌가요? 게스트 모드를 사용하여 비공개로 로그인하세요. 자세히 알아보기

계정 만들기 다음



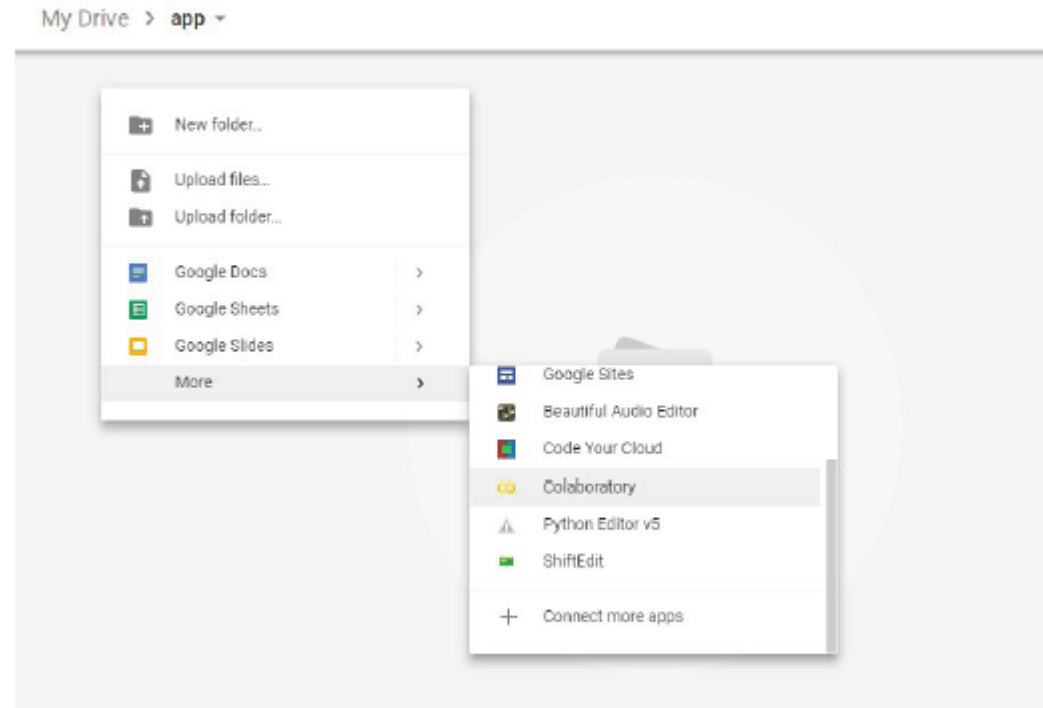
Google Colab 사용하기

- 새로운 App으로 Colaboratory 연결



I created an empty "app" folder

Create a new notebook via **Right click > More > Colaboratory**



Right click > More > Colaboratory

Google Colab 사용하기

- GPU 세팅

Notebook settings

Runtime type

Python 3



Hardware accelerator

GPU



☐ Omit code cell output when saving this notebook

CANCEL

SAVE

Google Colab 사용하기

- 구글 드라이브 마운트


```
1 from google.colab import drive
2 drive.mount('/content/drive/')
```

```
from google.colab import drive
drive.mount('/content/drive/')

... Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?c
Enter your authorization code:
|
```

```
from google.colab import drive
drive.mount('/content/drive/')

Go to this URL in a browser: https://accounts.google.com/o/oaut
Enter your authorization code:
.....
Mounted at /content/drive/
```



Q&A

Thank you for your listening

