

# Lab1. 파이썬 개발환경

*CS101 LAB*

# Announcement

## ■ Scoring policy

- 출석 5점 + Problem 5점 = 수업 한번당 10점
  - 단, 출석하지 않으면 무조건 0점.
  - 랩 숙제(problem)는 **다음날 23시 59분까지 제출**하면 제출한 것으로 인정된다.
- 지각
  - 랩 시작 전에 출석을 부른다. (출석 부를 때 없으면 지각으로 처리됨.)
  - 지각 2번은 결석 1번으로 처리된다.



아나콘다를 이용한 주피터 노트북 사용(윈도우 운영체제 기준)



# 아나콘다 설치

- Python 기반의 데이터 분석에 필요한 오픈소스를 모아놓은 개발 플랫폼
  - 파이썬 패키지 관리를 쉽게 해 주는 도구
  - 하나의 PC에 독립된 개발환경 구성 가능

- 설치      자신의 컴퓨터에 맞는 환경을 선택 후, 다운로드

- URL: <https://www.anaconda.com/download#downloads>

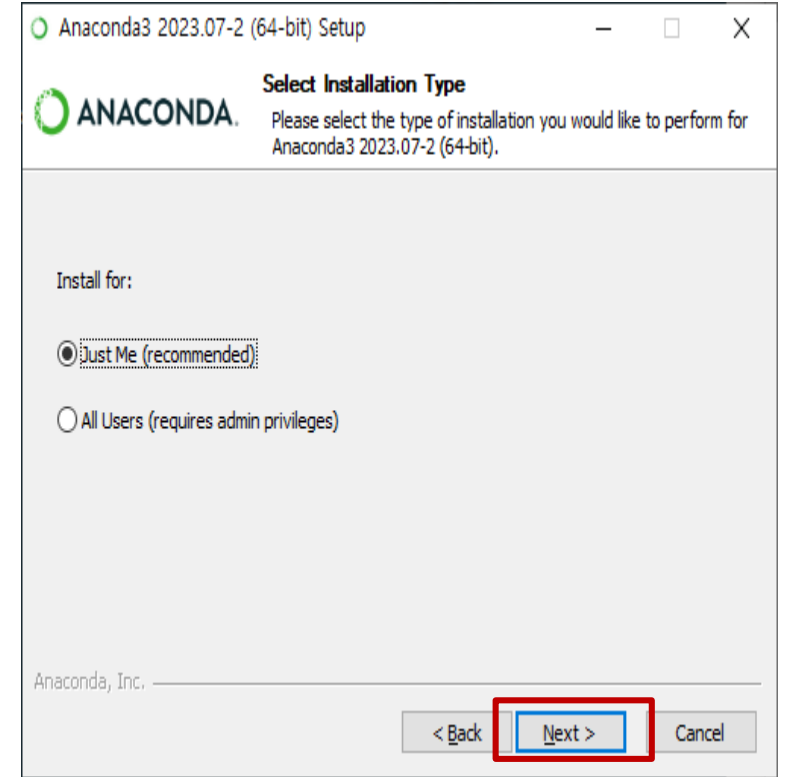
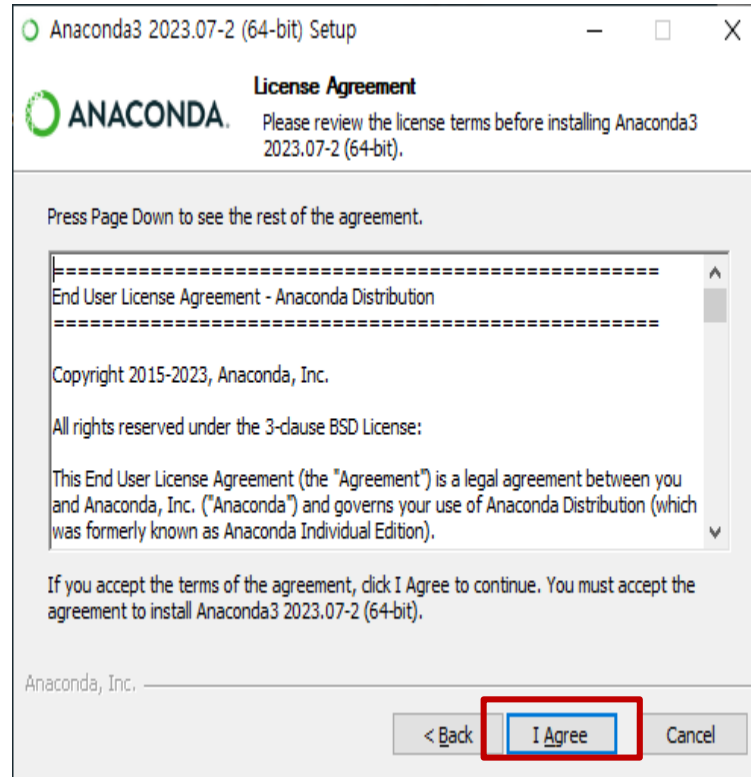
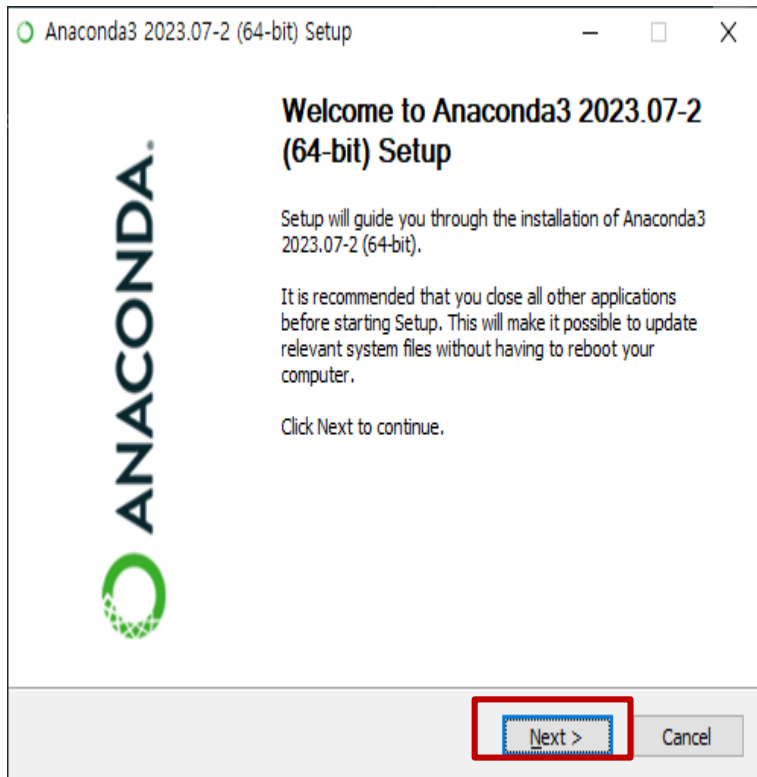


The screenshot shows the 'Anaconda Installers' page with three columns for Windows, Mac, and Linux. Each column lists download links for Python 3.11. The Windows column has a red box around the '64-Bit Graphical Installer (898.6 MB)' link. The Mac column has two links: '64-Bit Graphical Installer (610.5 MB)' and '64-Bit Command Line Installer (612.1 MB)'. The Linux column has two links: '64-Bit (x86) Installer (1015.6 MB)' and '64-Bit (Power8 and Power9) Installer (473.8 MB)'.

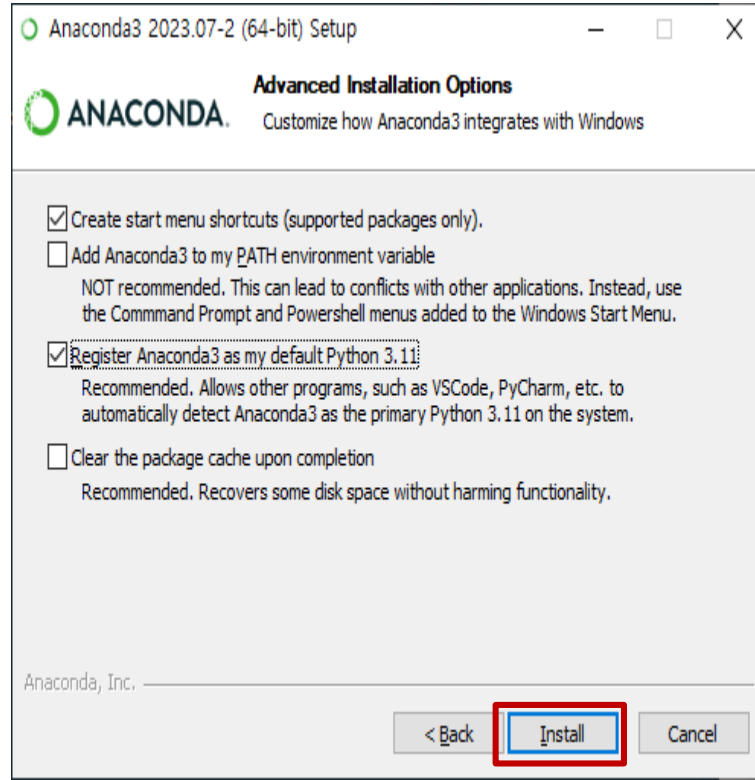
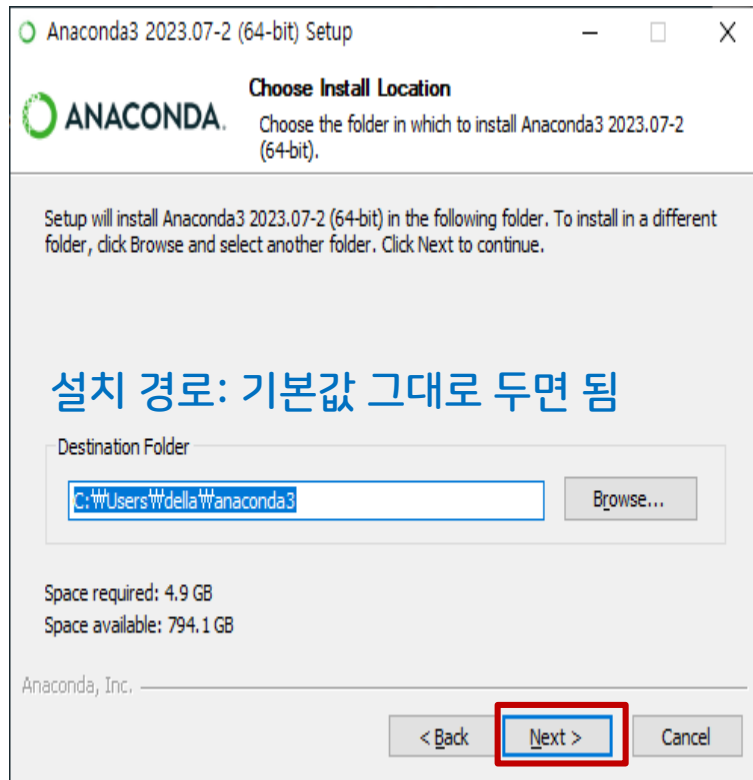
| Operating System | Python Version | Installer Type                       | Size      |
|------------------|----------------|--------------------------------------|-----------|
| Windows          | Python 3.11    | 64-Bit Graphical Installer           | 898.6 MB  |
|                  |                | 64-Bit Command Line Installer        | 612.1 MB  |
| Mac              | Python 3.11    | 64-Bit Graphical Installer           | 610.5 MB  |
|                  |                | 64-Bit Command Line Installer        | 612.1 MB  |
| Linux            | Python 3.11    | 64-Bit (x86) Installer               | 1015.6 MB |
|                  |                | 64-Bit (Power8 and Power9) Installer | 473.8 MB  |

# 아나콘다 설치

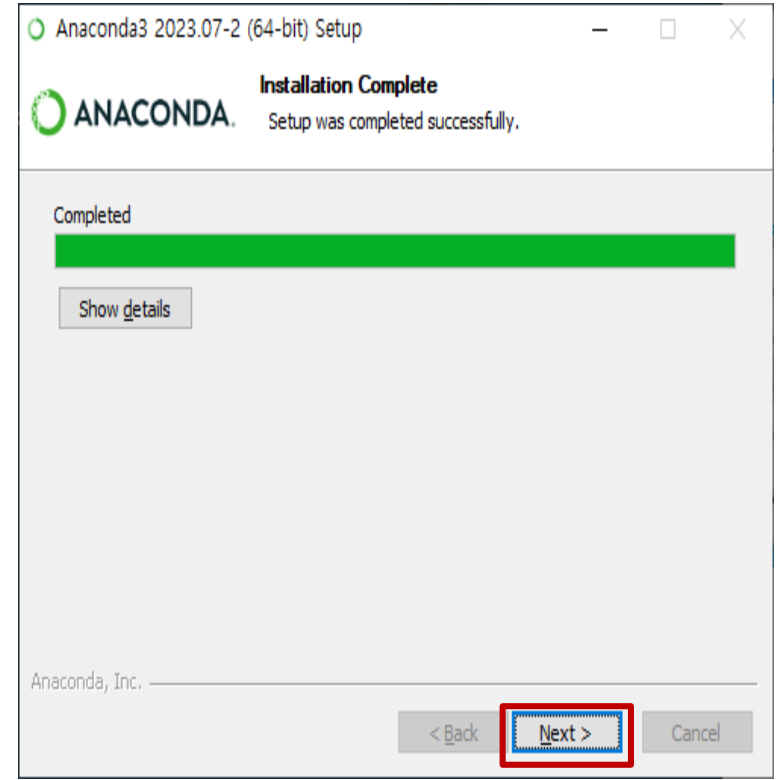
- 다운로드 한 Anaconda3-2023.07-2-Windows-x86\_64.exe 파일 실행  
버전은 변경될 수 있으며 다운로드 받은 버전으로 설치하면 됨



# 아나콘다 설치

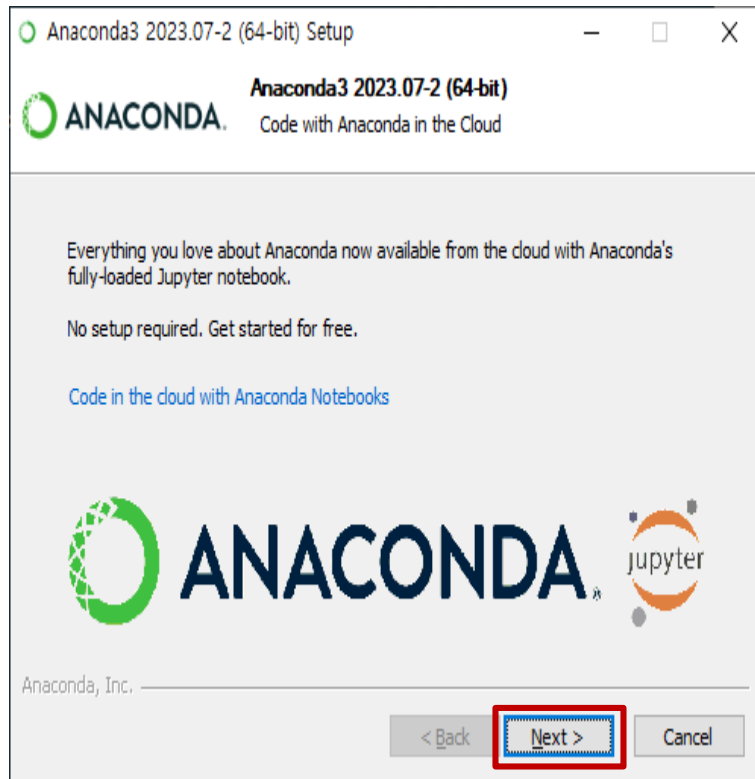


## [설치 완료]



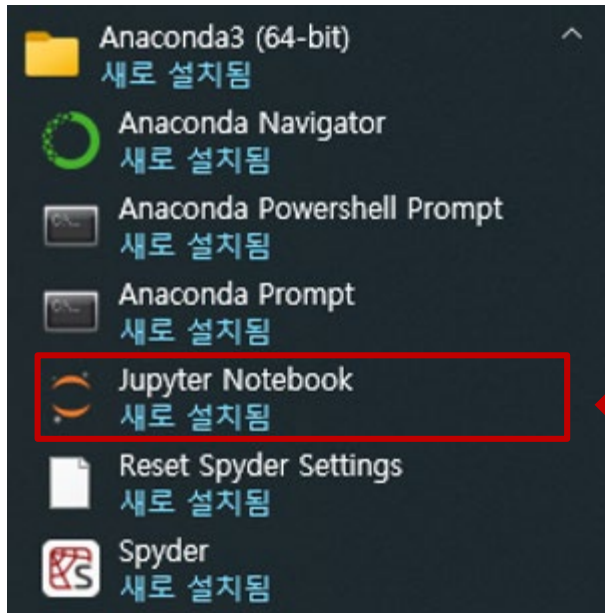
※ 설치 경로에 한글(또는 띄어쓰기)이 있으면 안됨  
이 경우, C드라이브 아래 anacond3라는 폴더 생성 후 해당 경로에 설치할 것

# 아나콘다 설치

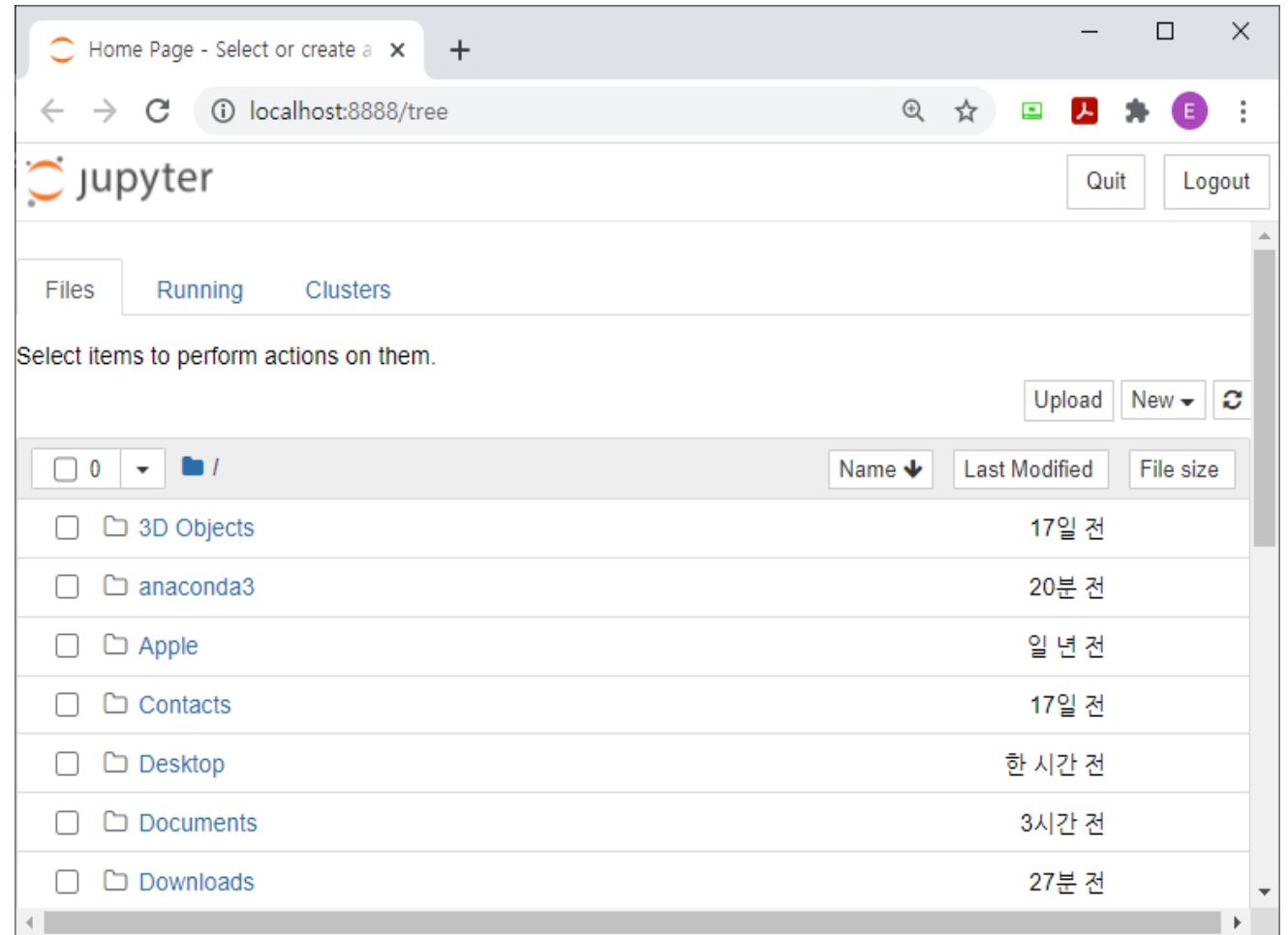


# Jupyter notebook 실행

- 아나콘다가 정상 설치되었다면, [시작]-[앱 목록]-[Anaconda3]에서 확인할 수 있다.



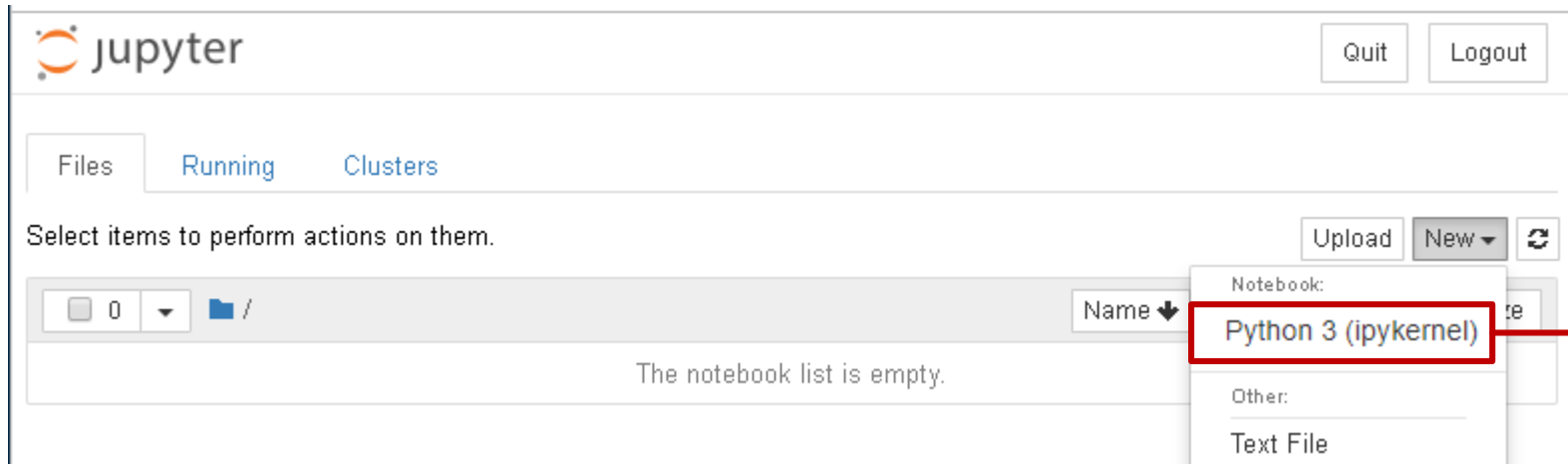
선택하여 실행



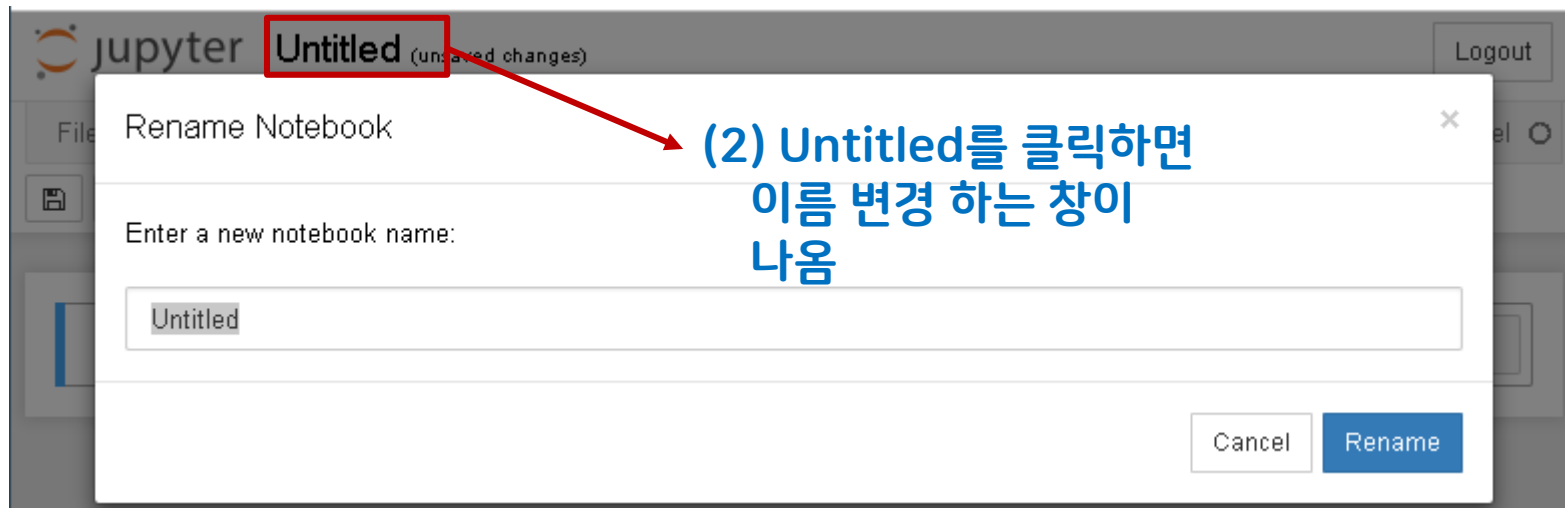
C:\Users\User\_id 를 기준으로 구조가 보임



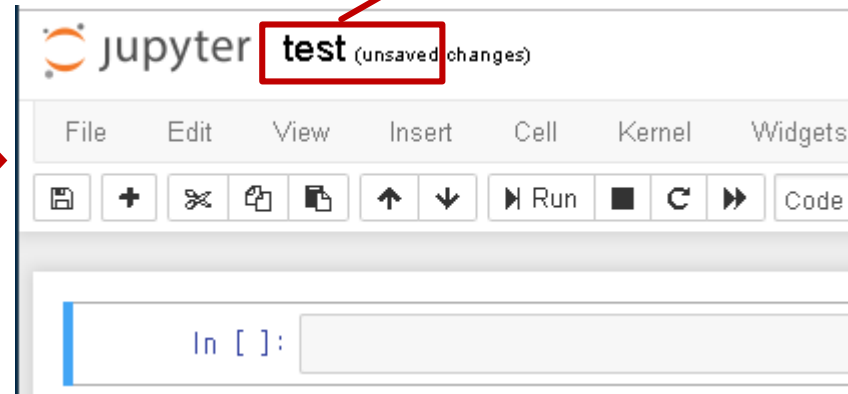
# Jupyter notebook 실행



(1) New → Python 3 선택



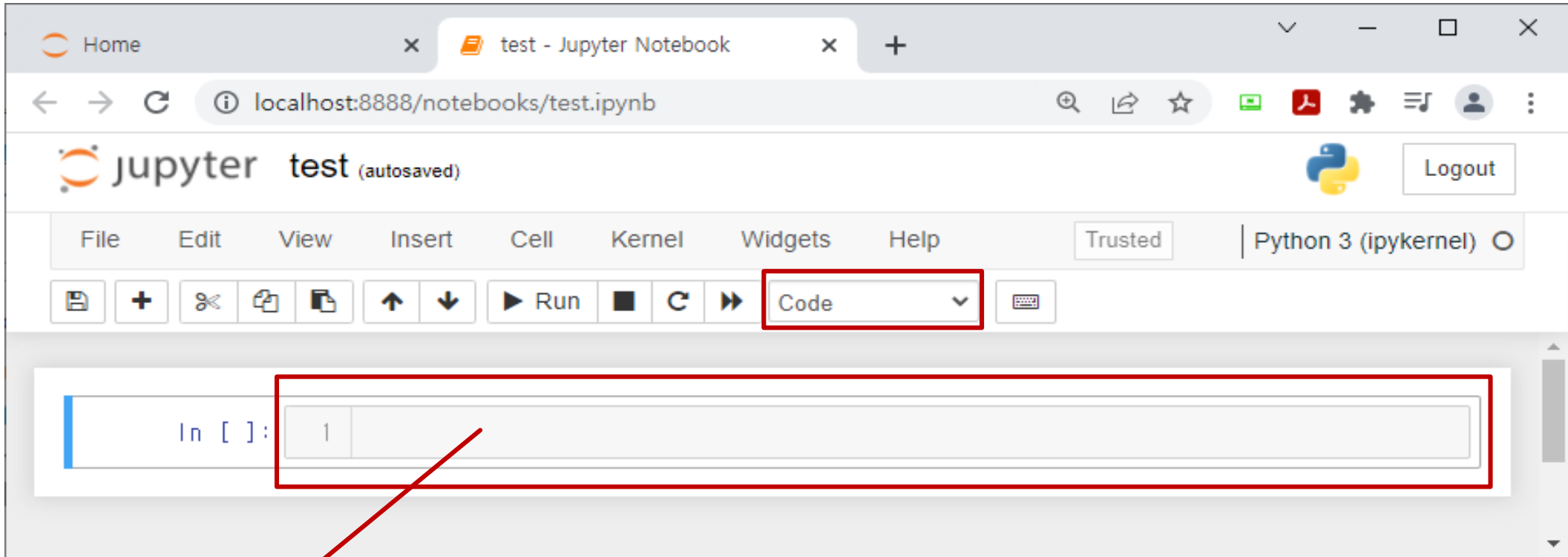
(2) Untitled를 클릭하면  
이름 변경 하는 창이  
나옴



(3) 이름 변경됨

# Cell type: Code

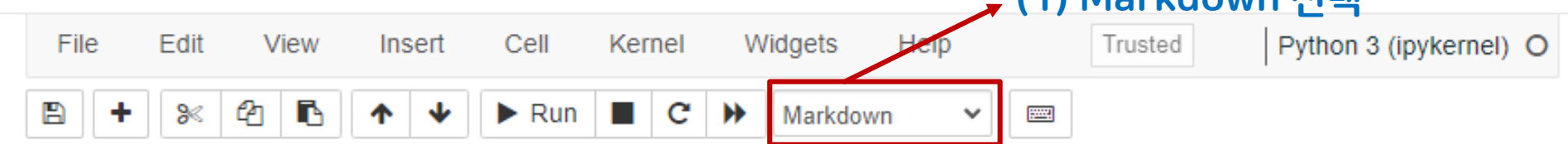
- Code는 Cell에 Python 코드를 입력하는 모드



셀

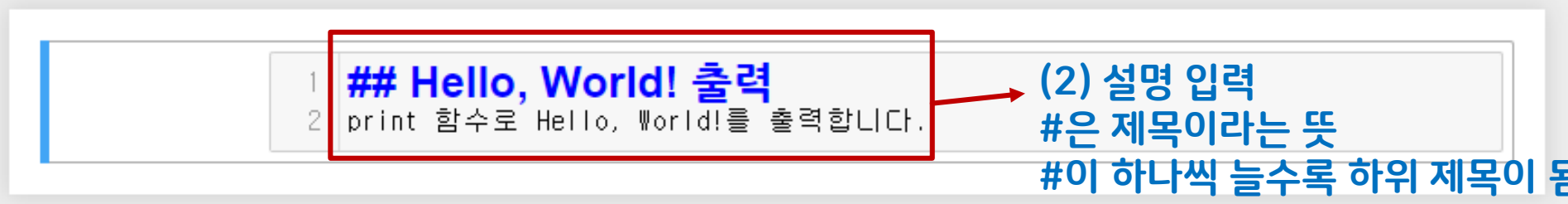
# Cell type: Markdown

(1) Markdown 선택



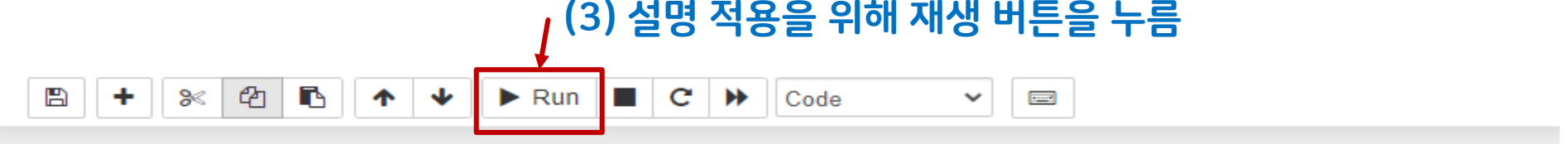
The screenshot shows the JupyterLab top bar with the 'Cell' menu open. The 'Markdown' option is highlighted with a red box and a red arrow pointing to it from the text '(1) Markdown 선택'.

(2) 설명 입력  
#은 제목이라는 뜻  
#이 하나씩 늘수록 하위 제목이 됨



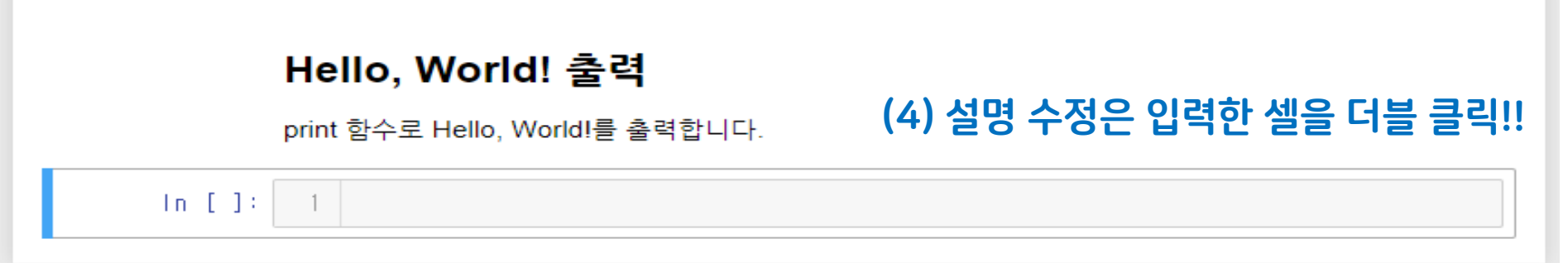
The screenshot shows a JupyterLab cell with a red box around the input text. The text is: `1 ## Hello, World! 출력`  
`2 print 함수로 Hello, World!를 출력합니다.` A red arrow points from the text '(2) 설명 입력' to the first line of the cell.

(3) 설명 적용을 위해 재생 버튼을 누름



The screenshot shows the JupyterLab top bar with the 'Run' button highlighted by a red box and a red arrow pointing to it from the text '(3) 설명 적용을 위해 재생 버튼을 누름'.

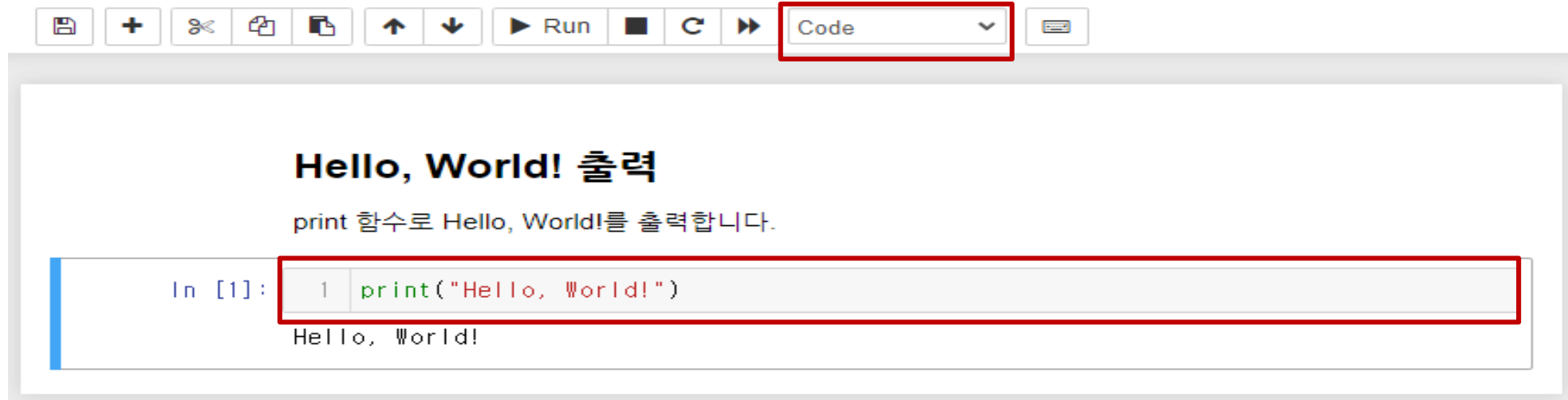
(4) 설명 수정은 입력한 셀을 더블 클릭!!



The screenshot shows the JupyterLab interface after running the cell. The output is displayed as: **Hello, World! 출력**  
print 함수로 Hello, World!를 출력합니다.  
Below the output is an input area with 'In [ ]:' and a text box containing '1'. A red arrow points from the text '(4) 설명 수정은 입력한 셀을 더블 클릭!!' to the input area.

# 파이썬 코드 입력 및 실행

- Cell에 아래와 같이 한 줄 코드를 입력한다.



- 실행

(1) 툴바 아이콘에서 재생버튼을 누름  
(2) 또는, Ctrl + enter를 입력  
코드가 실행되고 그 아래에 결과가 표시된다.

※ 주피터 노트북은 코드, 코드의 실행 결과, 코드에 대한 설명을 한 번에 저장할 수 있어 체계적인 기록이 가능

# 단축키

- Ctrl + enter

Cell에 표현식을 입력하고 ctrl + enter 입력 시, 현재 셀이 실행됨

- Shift + enter

현재 셀 실행되고, 다음 셀이 활성화

- Alt + enter

현재 셀 실행되고, 아래 셀 삽입

- Ctrl + s

현재 jupyter notebook에 저장되었다는 표시가 남음

# 단축키

- 들어쓰기: Tab
- 내어쓰기: Shift + Tab
- 주석처리, 주석풀기: Ctrl + /
- 줄번호: Cell 에서 Ctrl+m(또는 Esc)으로 명령(Command) 모드로 들어간 후에 L 을 치면 줄 번호가 toggle되어 보임



# Sample Program



# 산술 연산자

## ■ 산술 연산자

|              |                   |
|--------------|-------------------|
| $+, -, *, /$ | 더하기, 빼기, 곱하기, 나누기 |
| $\%$         | 나머지 (modulo)      |
| $//$         | 몫                 |
| $**$         | 제곱                |

※ 제곱 연산자(\*\*)는 다른 연산자들 보다 높은 우선 순위를 가짐

```
Anaconda Prompt (anaconda3) - python
>>> 2 + 4
6
>>> 210 - 91
119
>>> 3 * 12
36
>>> 7 / 2
3.5
>>>
```

## ■ 실습하기

(1)  $7 \% 3$   
(2)  $7 // 3$   
(3)  $2 ** 3$

(4)  $2 ** 4 + 5$   
(5)  $3 * 5 ** 2$



# 변수

## ■ 변수(variable)

- 어떤 값을 저장하기 위한 메모리 공간

## ■ 변수 이름 규칙

- 변수 이름은 알파벳, 숫자, underline( \_ )로 구성되는데, 첫 글자는 반드시 알파벳 또는 underline( \_ )로 시작함
- 이름 중간에 공백이 올 수 없음
- 대소문자 구분함
- 이미 사용하고 있는 몇몇 예약어(if, for, return 등)는 사용할 수 없음
- 내장 함수 이름이나 모듈 이름은 피할 것



```
>>>import keyword  
>>>keyword.kwlist      # 예약어 목록을 보여줌
```

# 변수와 기본 자료형

## ■ 변수 생성

- 파이썬은 변수에 값이 대입되는 시점에 변수의 자료형이 자동으로 정해짐

```
Anaconda Prompt (anaconda3) - python
>>> a = 100
>>> print(a)
100
>>>
```

※ = 는 대입연산자로 오른쪽의 것을 왼쪽으로 넣는다라는 의미

## ■ 기본 자료형

```
Anaconda Prompt (anaconda3) - python
>>> bool_var = True
>>> int_var = 100
>>> float_var = 123.45
>>> str_var = "python"
>>> type(bool_var), type(int_var), type(float_var), type(str_var)
(<class 'bool'>, <class 'int'>, <class 'float'>, <class 'str'>)
```

※ type() 함수는 변수의 종류를 확인하는 함수

# input()를 사용한 입력

## ■ input() 사용법

변수

사용자가 입력한 문자열을 정수로 변환한다

```
number = int(input("첫 번째 숫자를 입력하세요 : "))
```

안내 메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 받는다

선택 Anaconda Prompt (anaconda3) - python

```
>>> name = input("Enter your name: ")
Enter your name: Hong Gildong
>>> print(name)
Hong Gildong
>>> type(name)
<class 'str'>
>>> _
```

# input()를 사용한 입력

## ■ 문자열을 정수로 변경

```
Anaconda Prompt (anaconda3) - python
>>> number = input("Enter a number: ")
Enter a number: 123
>>> print(number)
123
>>> type(number)
<class 'str'>
>>>
```



```
Anaconda Prompt (anaconda3) - python
>>> number = int(number)
>>> print(number)
123
>>> type(number)
<class 'int'>
>>>
```

### ※ 형 변환

- ※ 문자열 -> 정수: int() 함수 이용
- ※ 문자열 -> 실수: float() 함수 이용
- ※ 숫자를 문자열로 변환: str() 함수 사용

# print()를 사용한 출력

- print()

```
print("Hello Python")
```

- 변수와 같이 출력

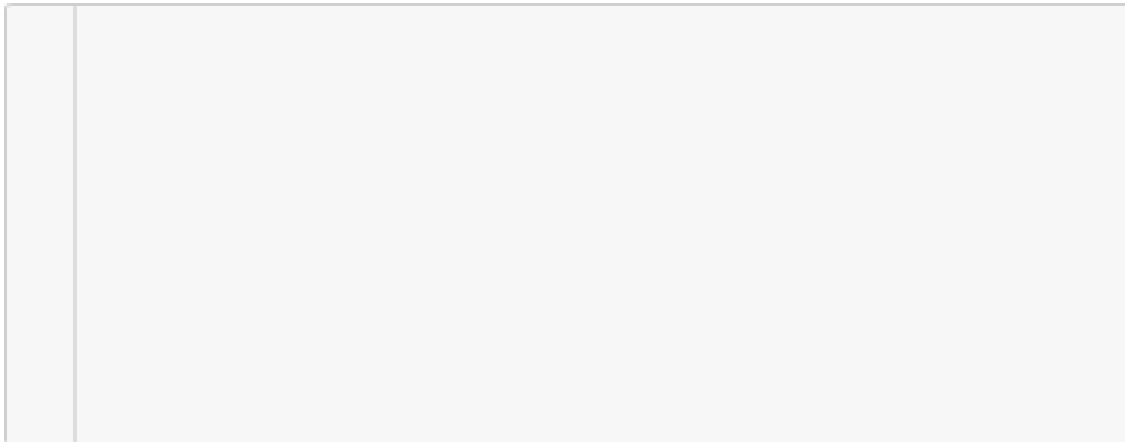
- 콤마(,) 사용

```
In [4]: 1 a = 100
        2 b = 200
        3 result = a + b
        4 print(a, "+", b, "=", result)
```

100 + 200 = 300

# Problem

- 두 개의 정수를 입력 받아 사칙 연산의 결과를 아래와 같이 출력하는 프로그램을 작성하시오.



```
첫 번째 정수를 입력하세요: 10  
두 번째 정수를 입력하세요: 5  
10 + 5 = 15  
10 - 5 = 5  
10 * 5 = 50  
10 / 5 = 2.0
```

- 제출 파일명: 학번\_Lab1.py