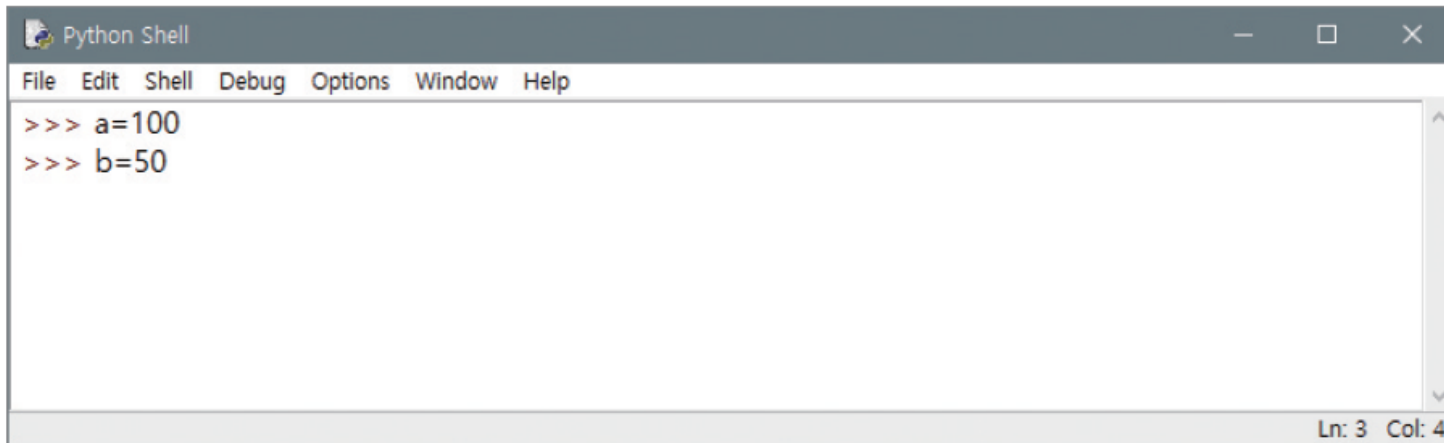


**미리 만드느 쓸 만한 프로그램**

# 계산기 프로그램의 기본 기능 구현

## ❖ 필요한 변수 준비

- = 기호는 같다는 의미가 아니라 ‘오른쪽의 것을 왼쪽으로 넣어라’는 의미의 대입 연산자
  - $a=100$ 은  $a \leftarrow 100$ 과 같은 개념
  - 내부적으로는 a 와 b 그릇이 생겨 a 그릇에는 100이, b 그릇에는 50이 담긴 상태
- 프로그래밍 언어에서 그릇과 같은 역할을 하는 것이 바로 변수

A screenshot of a Python Shell window. The title bar says 'Python Shell'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows two lines of code: '>>> a=100' and '>>> b=50'. The status bar at the bottom right indicates 'Ln: 3 Col: 4'.

그릇 이름 : a

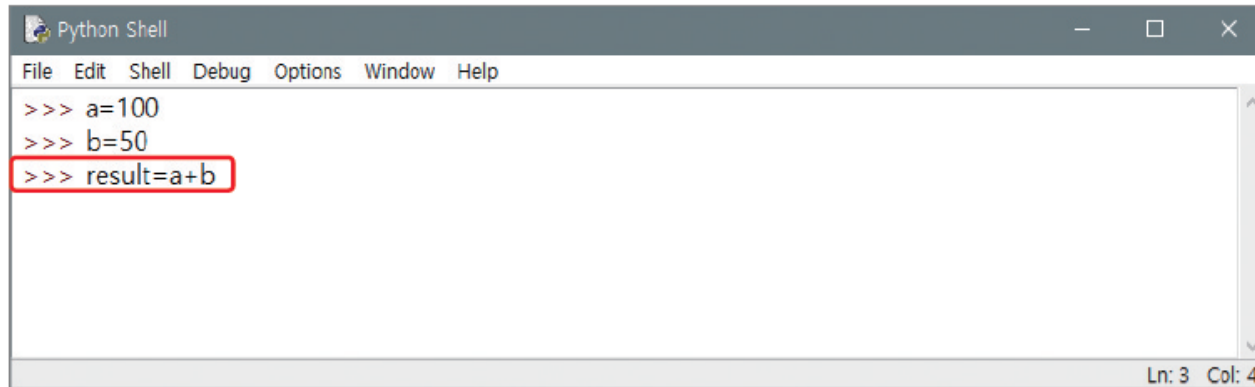


그릇 이름 : b

# 계산기 프로그램의 기본 기능 구현

## ❖ 더하기 기능 구현

- a 그릇의 100과 b 그릇의 50을 합쳐 새로운 result 그릇에 들어간 상태가 됨
- 변수는 result에 값이 들어가더라도 a, b의 변수값이 그대로 남음



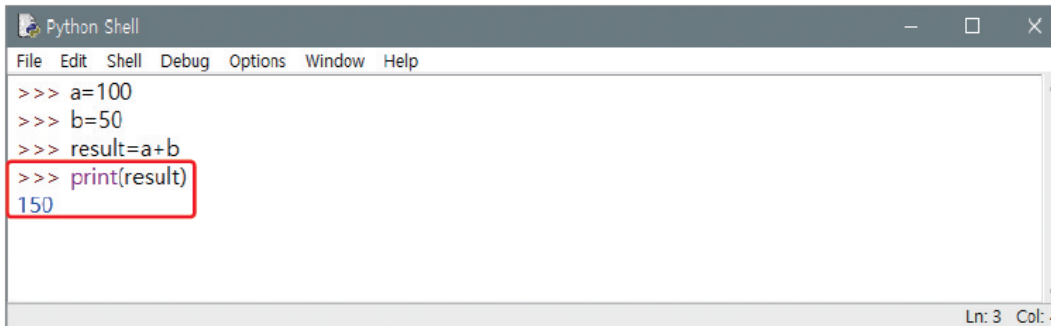
A screenshot of a Python Shell window. The window has a title bar 'Python Shell' and a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area contains three lines of code: `>>> a=100`, `>>> b=50`, and `>>> result=a+b`. The third line is highlighted with a red rectangular box. The status bar at the bottom right shows 'Ln: 3 Col: 4'.



# 계산기 프로그램의 기본 기능 구현

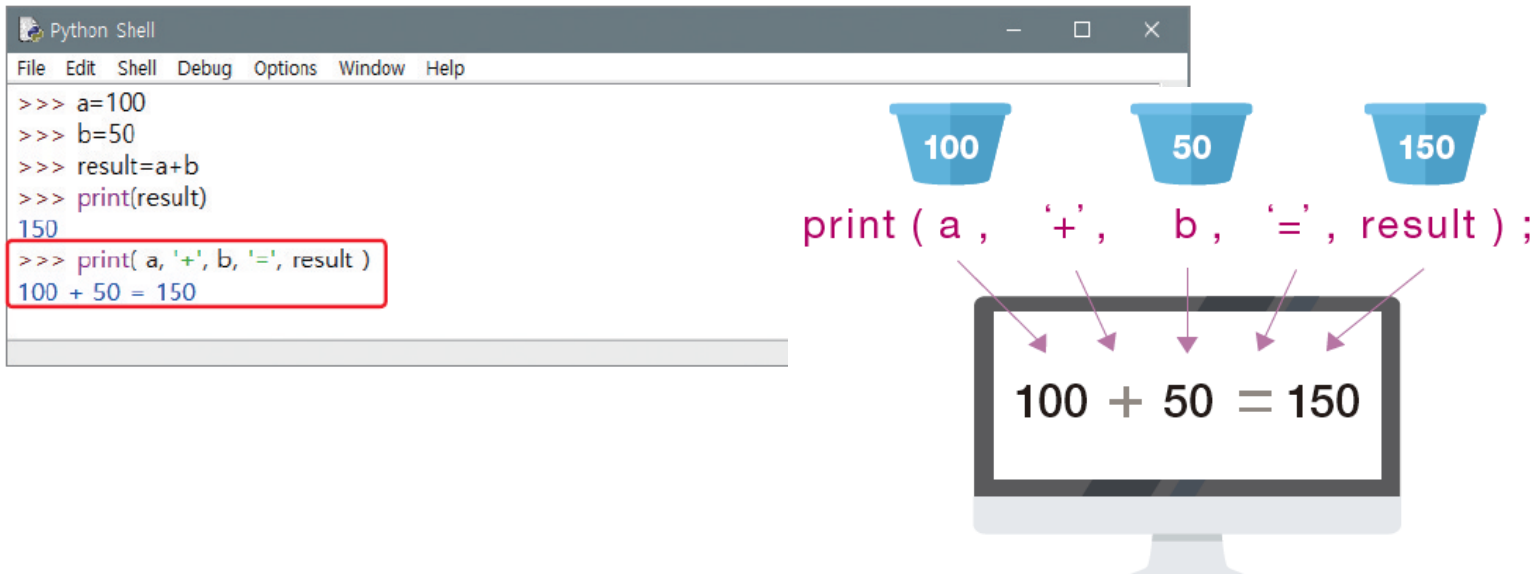
## ❖ 더한 결과 출력

➤ result 그릇의 내용만 출력



```
Python Shell
File Edit Shell Debug Options Window Help
>>> a=100
>>> b=50
>>> result=a+b
>>> print(result)
150
Ln: 3 Col: 4
```

➤ result 그릇의 내용과 계산식도 출력



```
Python Shell
File Edit Shell Debug Options Window Help
>>> a=100
>>> b=50
>>> result=a+b
>>> print(result)
150
>>> print( a, '+', b, '=', result )
100 + 50 = 150
```

The diagram illustrates the output of the second code block. Above the code, three blue buckets contain the values 100, 50, and 150. Arrows point from these buckets to the corresponding arguments in the `print` statement: `print ( a , '+' , b , '=' , result ) ;`. Below the code, a computer monitor displays the formatted output: `100 + 50 = 150`. Arrows also point from the variables `a`, `+`, `b`, `=`, and `result` in the code to the respective parts of the output on the monitor.

# 계산기 프로그램의 기본 기능 구현

## ❖ 함수

**함수(Function)**      함수는 특정 기능을 하도록 미리 만들어져 있거나 새로 만드는 것을 말한다. 함수는 뒤에 괄호가 붙는다. `print()`는 괄호 안의 내용을 출력해 함수로 파이썬이 미리 만들어서 제공하는 것이다. 이 외에도 파이썬은 매우 많은 함수를 제공하지만 프로그래머가 원하는 모든 함수를 파이썬에서 제공할 수는 없어 직접 만들어서 사용하기도 한다. 이 장 후반부에서 함수를 만드는 방법을 간단히 배우고, 더 자세한 내용은 9장에서 다룬다.

# 계산기 프로그램 저장

## ❖ 빼기, 곱하기, 나누기 기능 구현

A screenshot of a Python Shell window. The window has a title bar 'Python Shell' and a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main area shows a series of Python commands and their outputs. The commands are: `>>> a=100`, `>>> b=50`, `>>> result=a+b`, `>>> print(result)` (output: 150), `>>> print( a, '+', b, '=', result )` (output: 100 + 50 = 150), `>>> result=a-b`, `>>> print( a, '-', b, '=', result )` (output: 100 - 50 = 50), `>>> result=a*b`, `>>> print( a, '*', b, '=', result )` (output: 100 \* 50 = 5000), `>>> result=a/b`, `>>> print( a, '/', b, '=', result )` (output: 100 / 50 = 2.0). The last four lines of code and their outputs are enclosed in a red rectangular box. The status bar at the bottom right shows 'Ln: 19 Col: 4'.

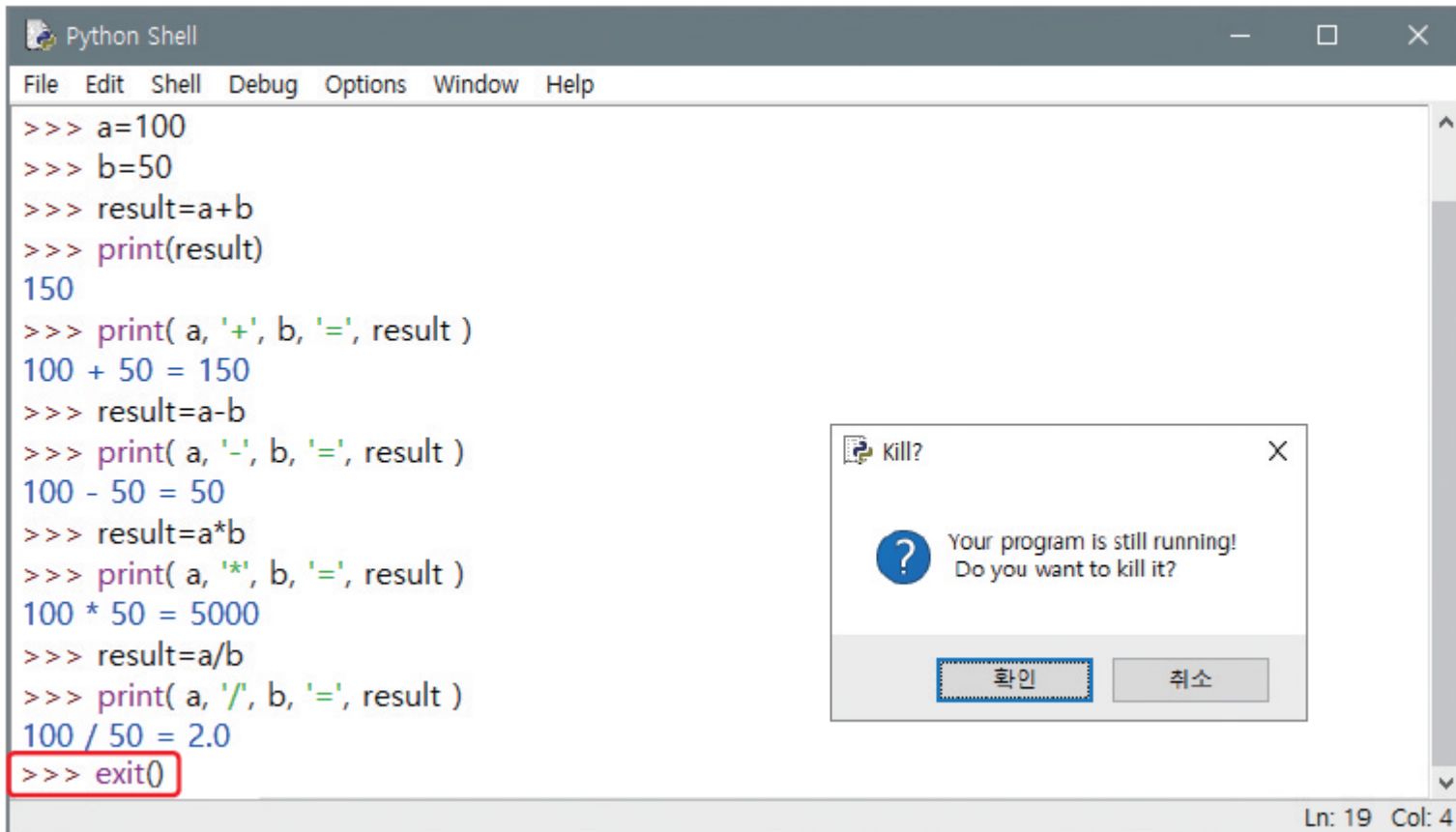
```
Python Shell
File Edit Shell Debug Options Window Help
>>> a=100
>>> b=50
>>> result=a+b
>>> print(result)
150
>>> print( a, '+', b, '=', result )
100 + 50 = 150
>>> result=a-b
>>> print( a, '-', b, '=', result )
100 - 50 = 50
>>> result=a*b
>>> print( a, '*', b, '=', result )
100 * 50 = 5000
>>> result=a/b
>>> print( a, '/', b, '=', result )
100 / 50 = 2.0
>>> |
```

Ln: 19 Col: 4

# 계산기 프로그램 저장

## ❖ IDLE 종료

- [File]-[Exit] 메뉴 선택
- `exit()` 코드를 입력한 후 “~Kill it?” 메시지창에서 <확인> 버튼 클릭



The screenshot shows a Python Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a command prompt. The following code has been executed:

```
>>> a=100
>>> b=50
>>> result=a+b
>>> print(result)
150
>>> print( a, '+', b, '=', result )
100 + 50 = 150
>>> result=a-b
>>> print( a, '-', b, '=', result )
100 - 50 = 50
>>> result=a*b
>>> print( a, '*', b, '=', result )
100 * 50 = 5000
>>> result=a/b
>>> print( a, '/', b, '=', result )
100 / 50 = 2.0
>>> exit()
```

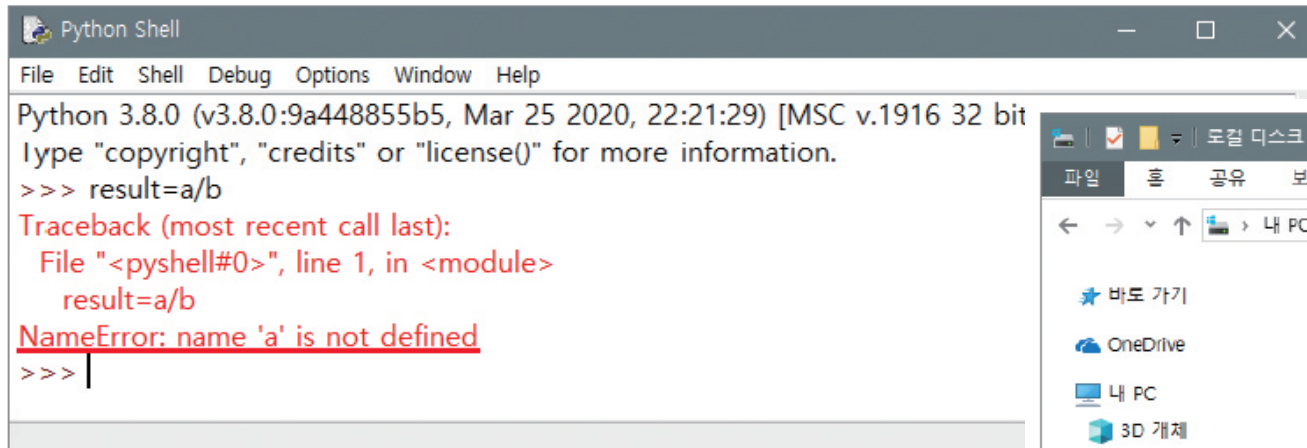
A dialog box titled "Kill?" is displayed in the foreground. It contains a question mark icon and the text: "Your program is still running! Do you want to kill it?". There are two buttons at the bottom: "확인" (Confirm) and "취소" (Cancel). The "확인" button is highlighted with a blue border.

Ln: 19 Col: 4

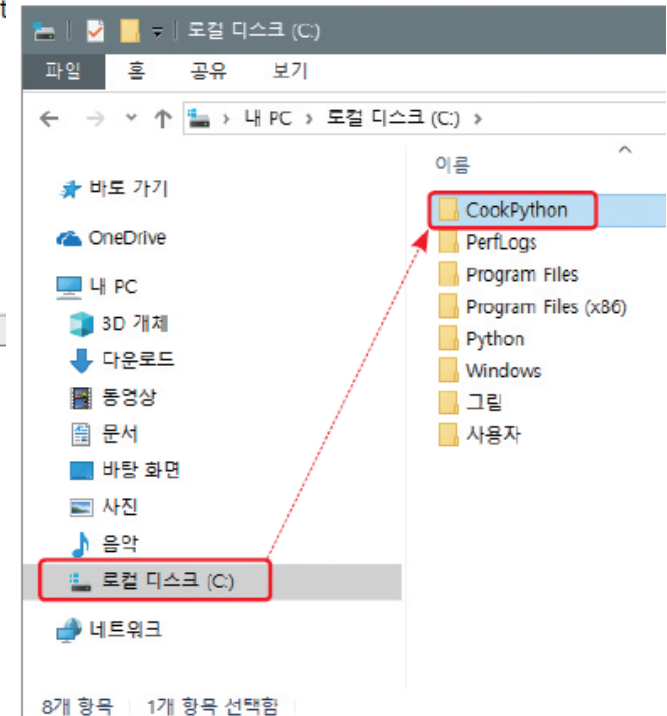
# 계산기 프로그램 저장

## ❖ 프로그램 저장의 필요성

- IDLE을 실행한 후 앞에서 입력한 나누기를 다시 실행
- 메모리에 저장된 것은 IDLE을 종료하면 모두 사라져 오류 발생
  - ✓ 처음부터 다시 입력해야 함
- 파이썬 코드는 하드디스크나 USB에 저장해 놓는 방법 사용



A screenshot of a Python Shell window. The title bar says "Python Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The text area shows the following content: "Python 3.8.0 (v3.8.0:9a448855b5, Mar 25 2020, 22:21:29) [MSC v.1916 32 bit type \"copyright\", \"credits\" or \"license()\" for more information. >>> result=a/b Traceback (most recent call last): File \"<pyshell#0>\", line 1, in <module> result=a/b NameError: name 'a' is not defined >>>". The error message "NameError: name 'a' is not defined" is underlined in red.





# 계산기 프로그램 저장

## ❖ 파이썬 파일 저장(1/2)

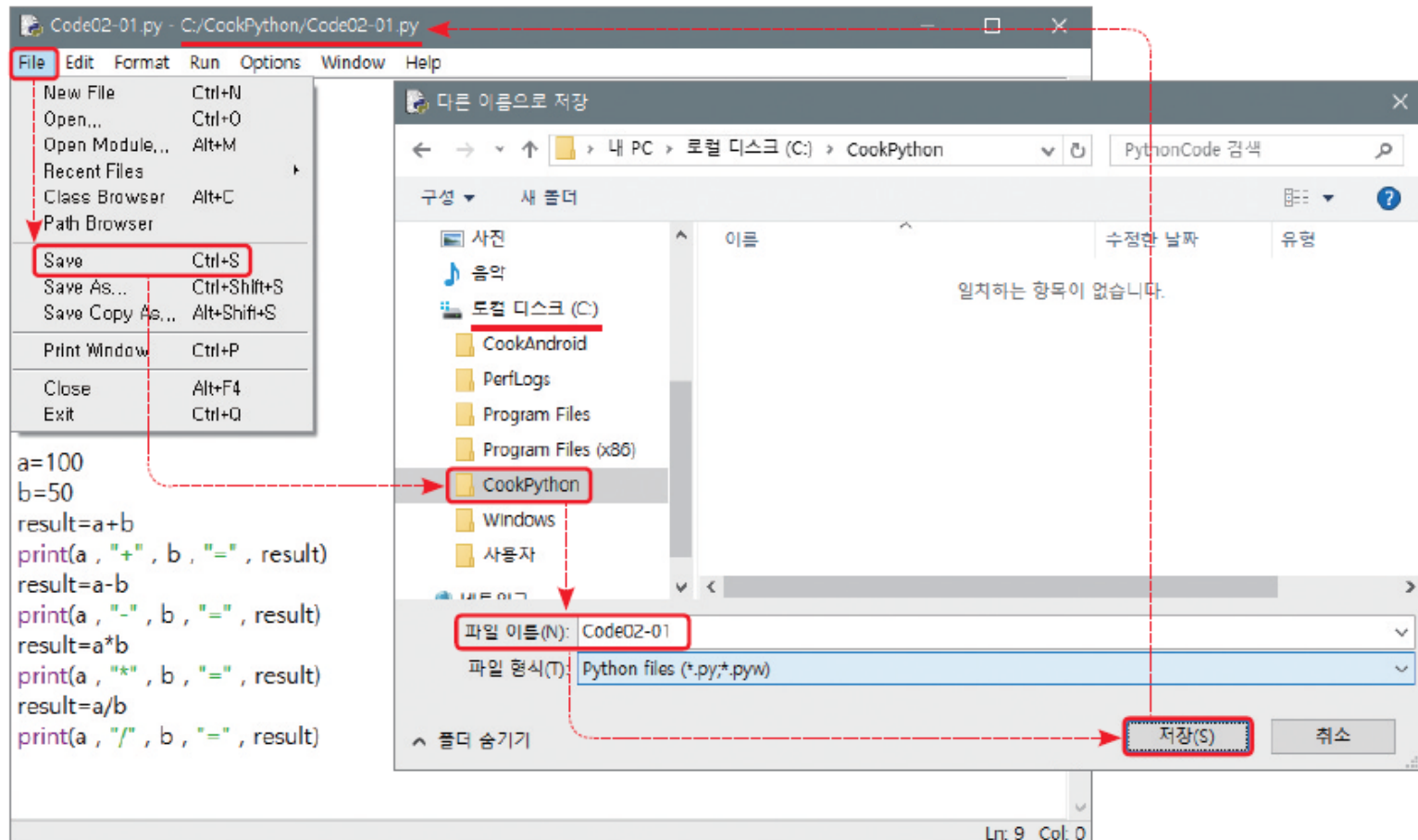
- 코드가 수십 줄인 경우는 스크립트 모드 사용
  - ✓ IDLE에서 [File]-[New File] 메뉴 선택
- 메모장 같은 창인 스크립트 모드에서 코드를 여러 줄 입력 가능
  - ✓ 단, 실행은 되지 않음

```
1 a = 100
2 b = 50
3 result = a + b
4 print(a, "+", b, "=", result)
5 result = a - b
6 print(a, "-", b, "=", result)
7 result = a * b
8 print(a, "*", b, "=", result)
9 result = a / b
10 print(a, "/", b, "=", result)
```

# 계산기 프로그램 저장

## ❖ 파이썬 파일 저장(2/2)

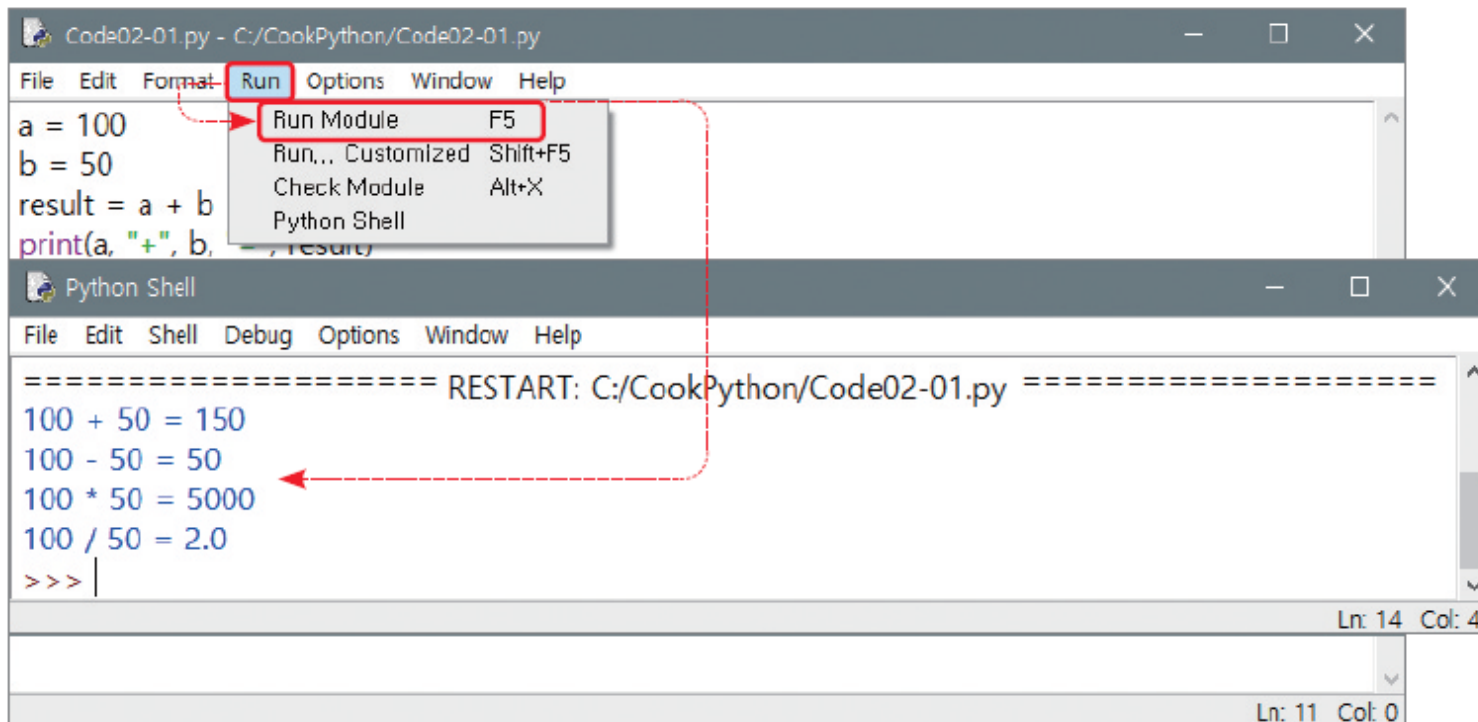
- 스크립트 모드에서 [File]-[Save] 메뉴를 선택해 특정 디렉토리(폴더)에 저장(확장명 \*.py가 자동으로 붙음)



# 계산기 프로그램 저장

## ❖ 파이썬 파일 실행

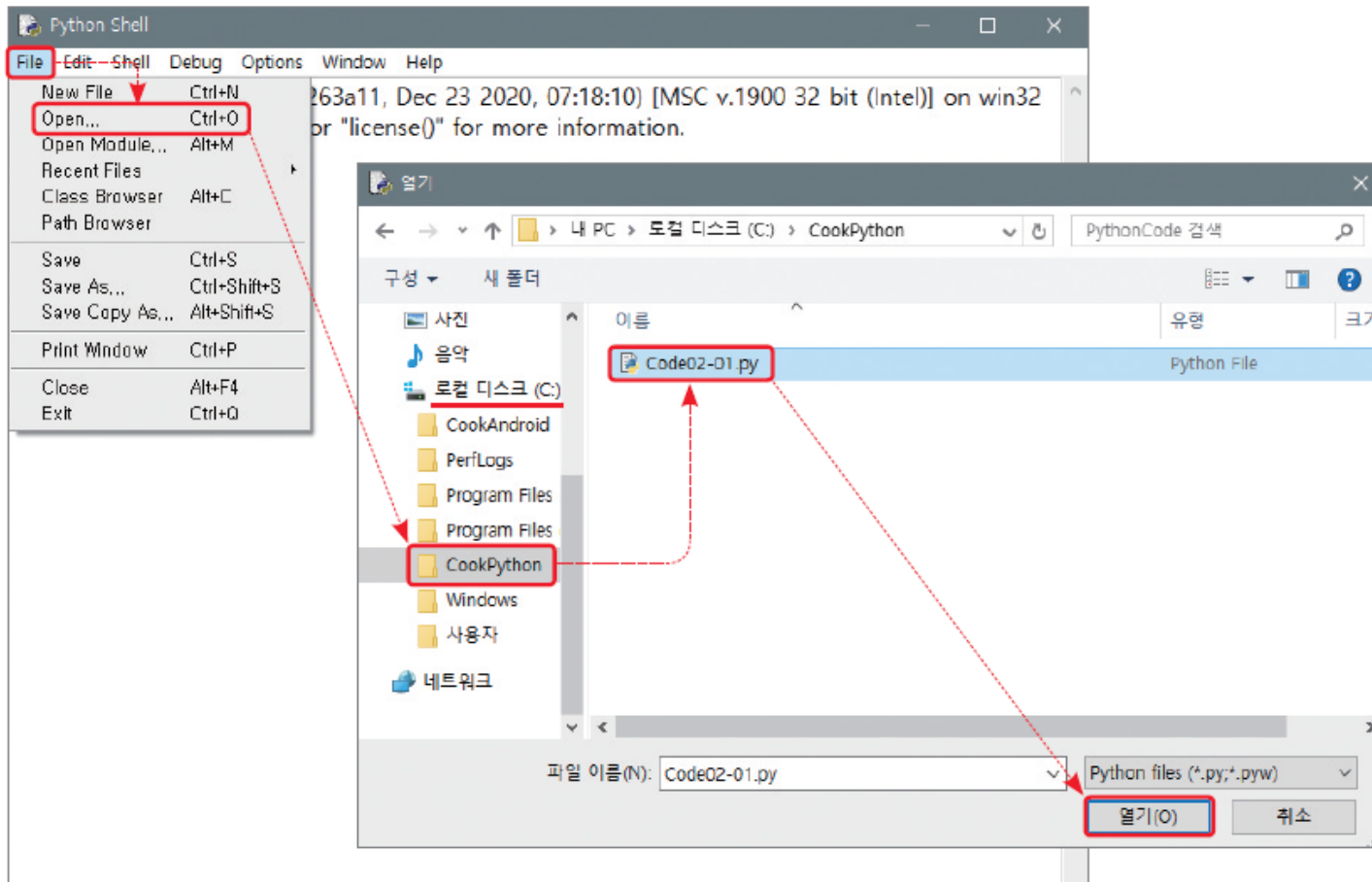
- 스크립트 모드에서 [Run]-[Run Module] 메뉴 선택(또는 [F5])



# 계산기 프로그램 저장

## ❖ 파이썬 파일 열기(1/2)

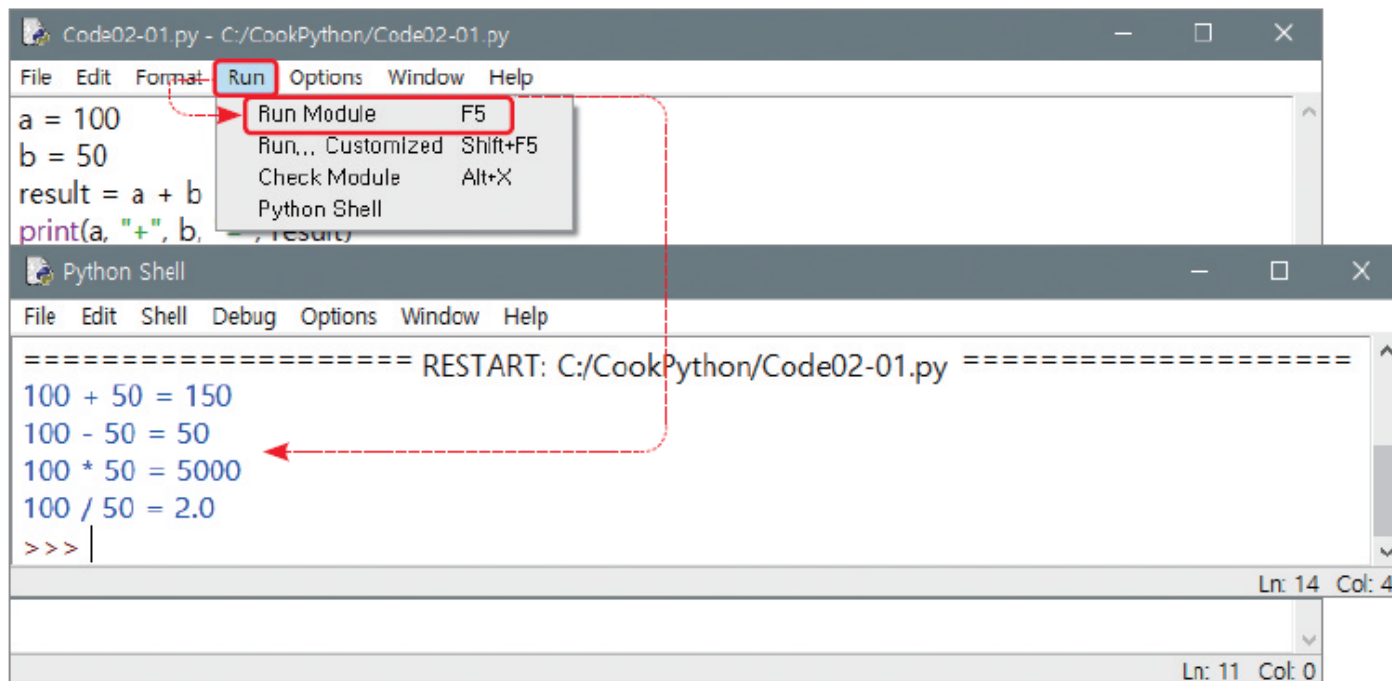
➢ IDLE에서 [File]-[Open] 메뉴를 선택한 후 저장된 파일 열기



# 계산기 프로그램 저장

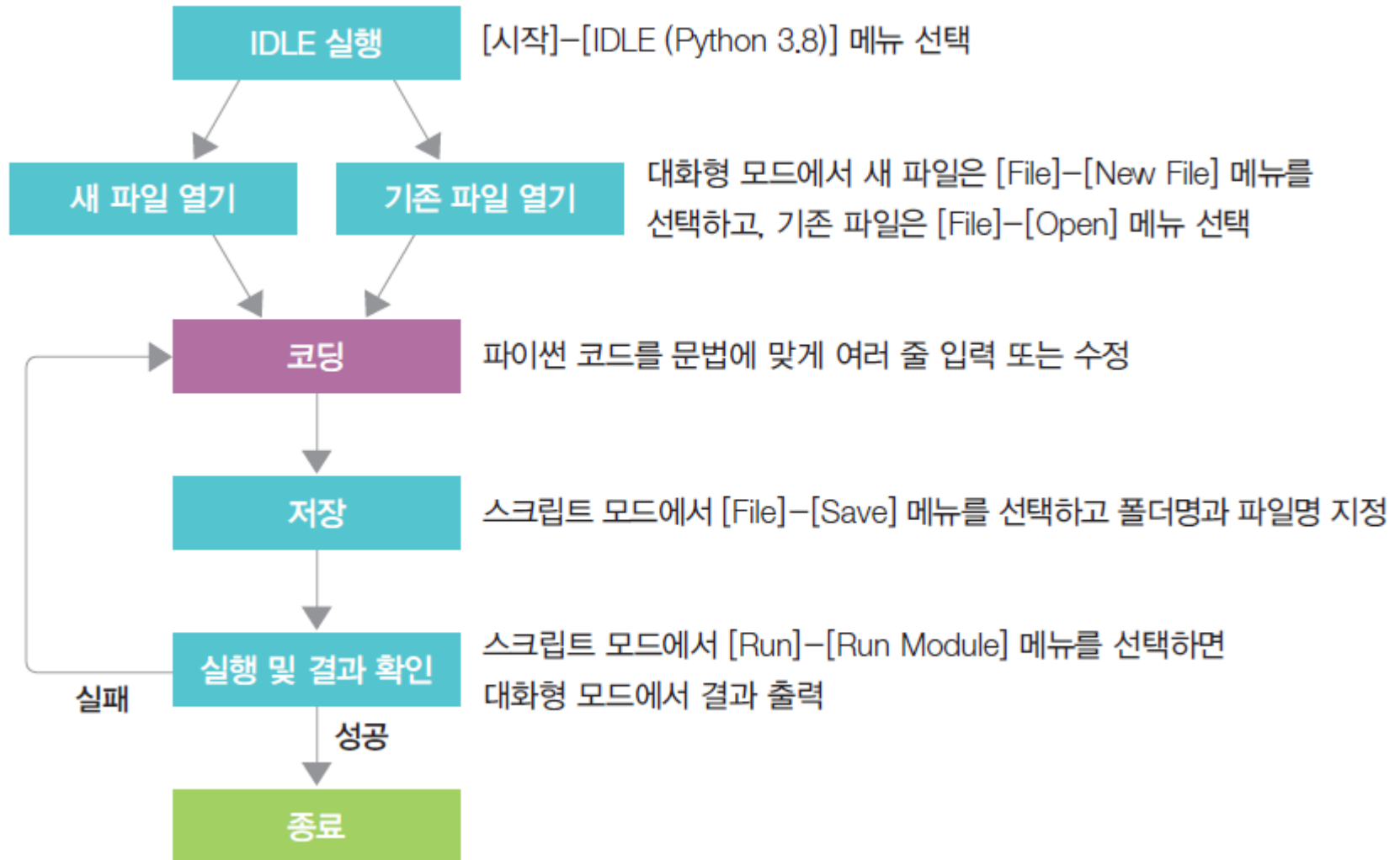
## ❖ 파이썬 파일 열기(2/2)

- IDLE에서 [File]-[Open] 메뉴를 선택한 후 파일을 열기
- a와 b의 값을 300과 200으로 수정한 후 [File]-[Save] 메뉴 선택(또는 [Ctrl] + [S])해 저장
- [Run]-[Run Module] 메뉴(또는 [F5])로 다시 실행



# 계산기 프로그램 저장

## ❖ 긴 프로그램을 코딩하는 순서



# 계산기 프로그램 확장

## ❖ 파이썬 파일 만들어 저장

- 100과 50을 고정적으로 계산하는 것이 아니라 직접 입력한 두 숫자의 사칙 연산을 수행하도록 프로그램 수정

변수 a에 100을 넣는다. → 변수 a에 입력할 값을 키보드로 입력받는다.  
변수 b에 50을 넣는다. → 변수 b에 입력할 값을 키보드로 입력받는다.



- 대화형 모드에서 [File]-[New File] 메뉴를 선택해 새 파일을 연 후 스크립트 모드에서 [File]-[Save] 메뉴를 선택해 작성된 프로그램을 저장

# 계산기 프로그램 확장

## ❖ input( ) 함수를 사용해 값 입력

- 1~2행을 input( ) 함수를 사용하도록 수정하고 [F5]를 눌러 실행

```
1 a = input()
2 b = input()
3 result = a + b
4 print(a, "+", b, "=", result)
5 result = a - b
6 print(a, "-", b, "=", result)
7 result = a * b
8 print(a, "*", b, "=", result)
9 result = a / b
10 print(a, "/", b, "=", result)
```

- 숫자 하나를 입력하고 [Enter] → 다시 숫자 하나를 입력하고 [Enter]

```
Python Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/CookPython/Code02-02.py =====
100
50
100 + 50 = 10050
Traceback (most recent call last):
  File "C:/CookPython/Code02-02.py", line 5, in <module>
    result=a-b
TypeError: unsupported operand type(s) for -: 'str' and 'str'
>>> |
```

계산 결과가 틀리거나 오류 발생  
input() 함수는 값을 입력받지만 모두 문자열로  
취급하기 때문



# 계산기 프로그램 확장

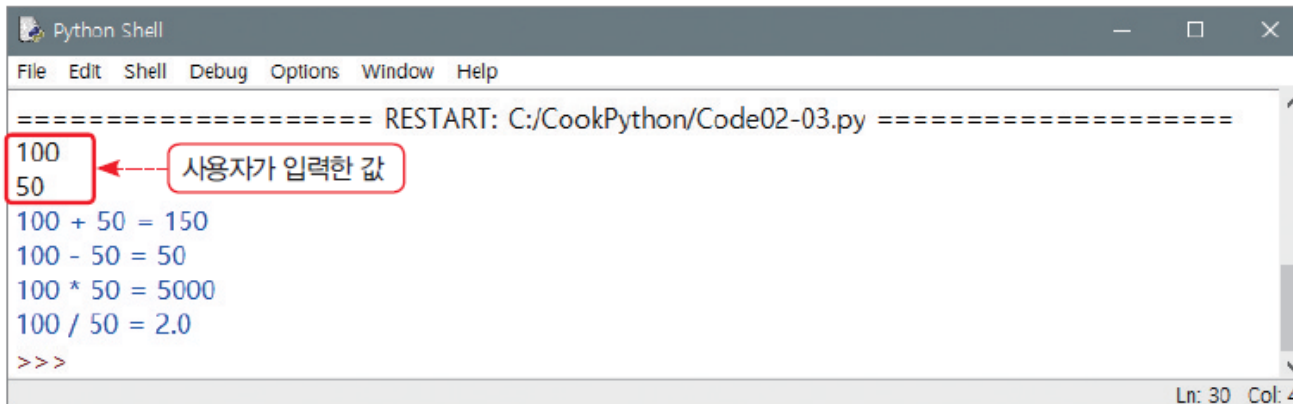
## ❖ input( ) 함수를 사용해 정수로 변환

- 입력된 값을 int() 함수를 사용해 정수로 변환

```
int("100")      # 결과는 정수 100  
int(100.123)    # 결과는 정수 100
```

- 1~2행을 다음과 같이 수정 후 다시 [F5]를 눌러 실행

```
1 a = int(input())  
2 b = int(input())
```

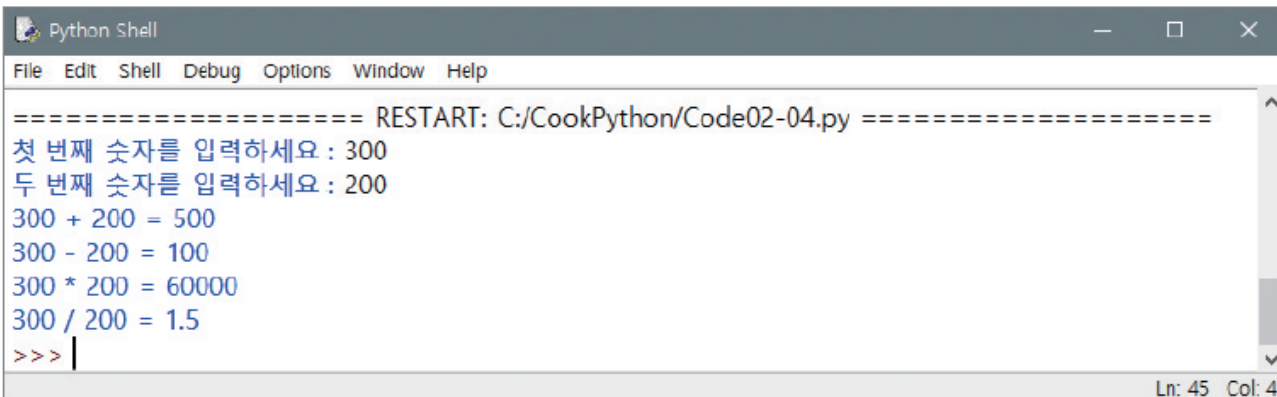


```
Python Shell  
File Edit Shell Debug Options Window Help  
===== RESTART: C:/CookPython/Code02-03.py =====  
100  
50 ← 사용자 입력한 값  
100 + 50 = 150  
100 - 50 = 50  
100 * 50 = 5000  
100 / 50 = 2.0  
>>>  
Ln: 30 Col: 4
```

# 계산기 프로그램 확장

## ❖ 계산기의 최종 버전

```
1 a = int(input("첫 번째 숫자를 입력하세요 : "))
2 b = int(input("두 번째 숫자를 입력하세요 : "))
3 result = a + b
4 print(a, "+", b, "=", result)
5 result = a - b
6 print(a, "-", b, "=", result)
7 result = a * b
8 print(a, "*", b, "=", result)
9 result = a / b
10 print(a, "/", b, "=", result)
```



The screenshot shows a Python Shell window titled "Python Shell" with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The main text area displays the output of the program after a restart. It shows two input prompts: "첫 번째 숫자를 입력하세요 : 300" and "두 번째 숫자를 입력하세요 : 200". Below these, the results of the four operations are displayed: "300 + 200 = 500", "300 - 200 = 100", "300 \* 200 = 60000", and "300 / 200 = 1.5". The prompt ">>>" is visible at the bottom left. The status bar at the bottom right indicates "Ln: 45 Col: 4".

```
Python Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/CookPython/Code02-04.py =====
첫 번째 숫자를 입력하세요 : 300
두 번째 숫자를 입력하세요 : 200
300 + 200 = 500
300 - 200 = 100
300 * 200 = 60000
300 / 200 = 1.5
>>> |
Ln: 45 Col: 4
```