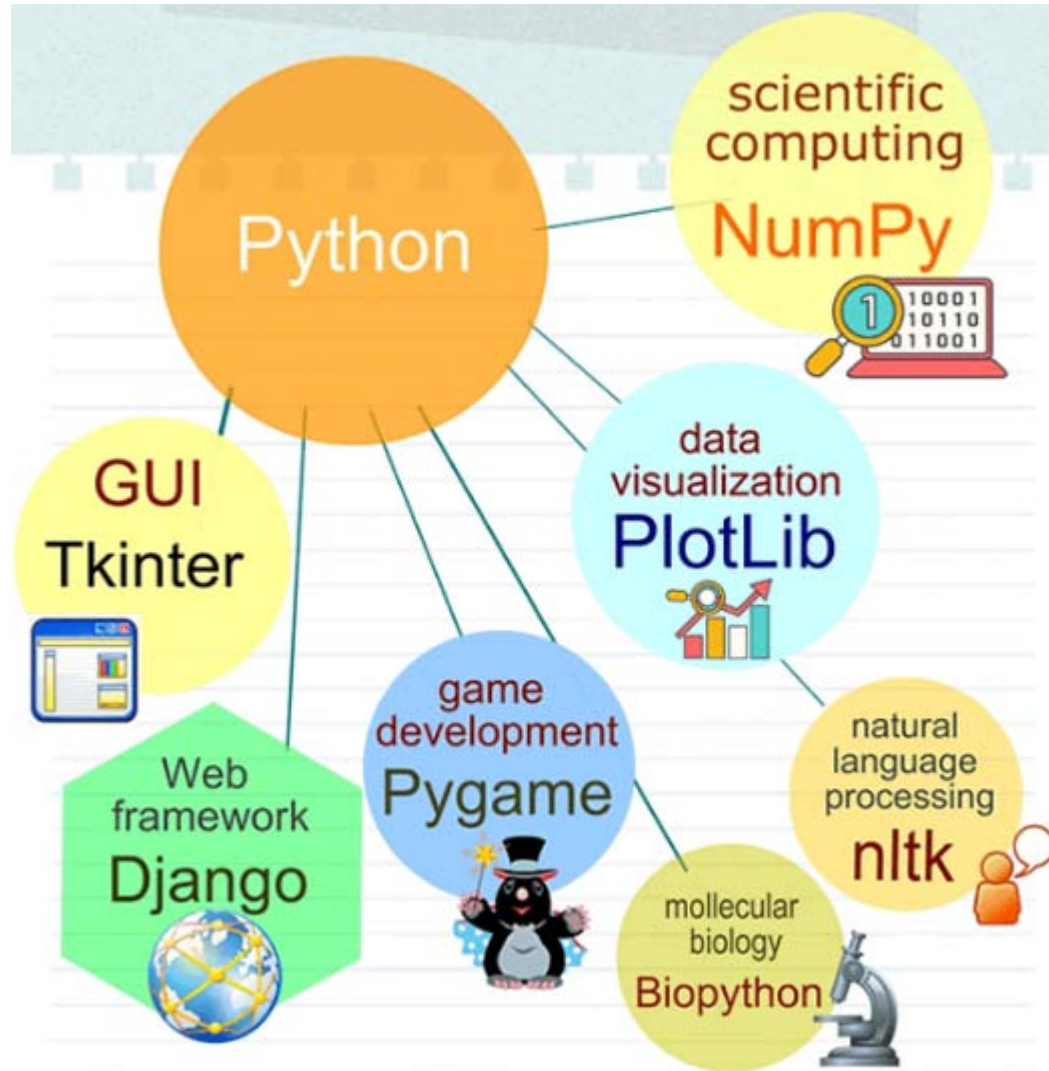


파이썬 소개 및 설치방법

파이썬 (Python)

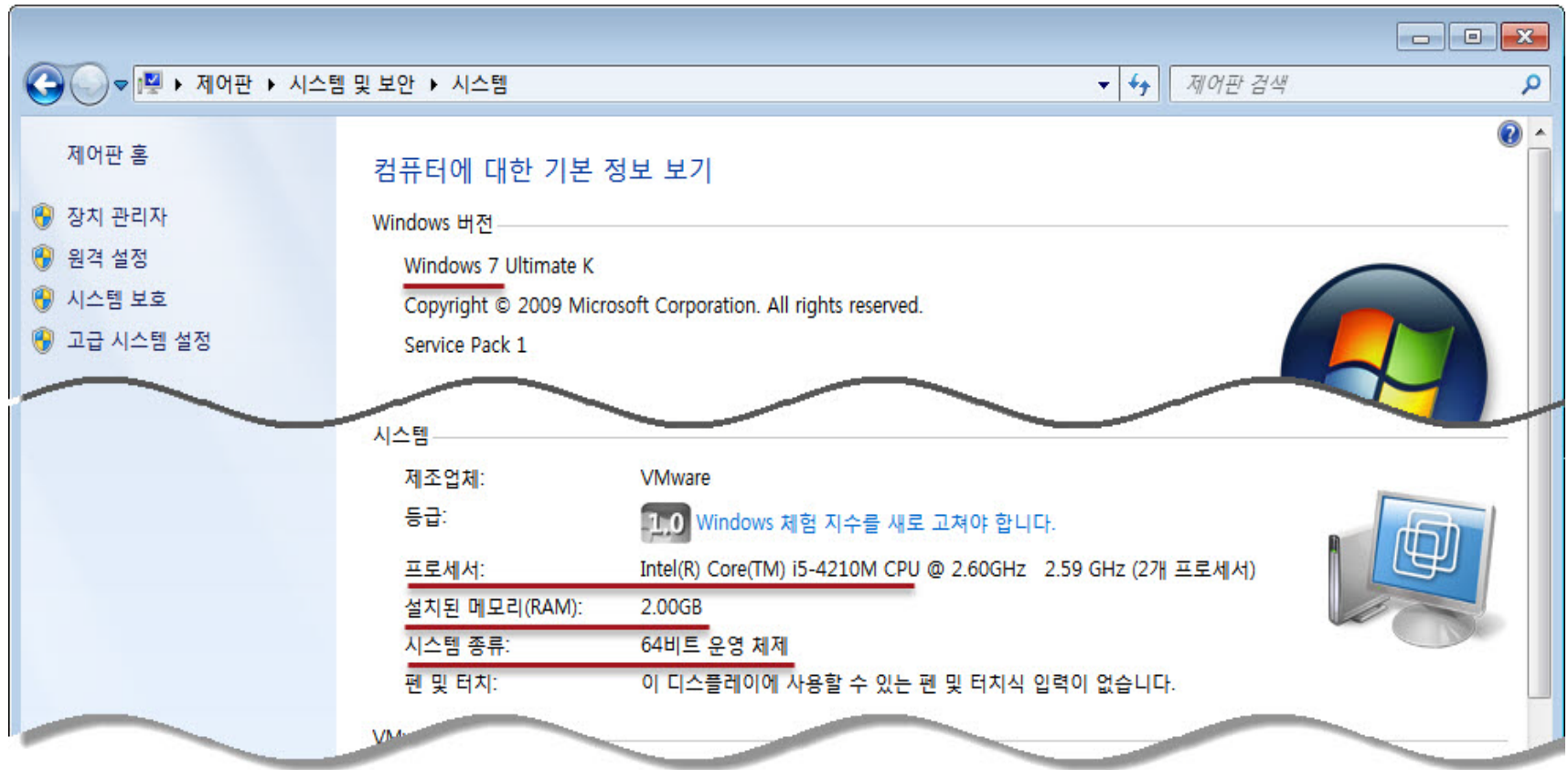
- ❖ 1991년에 네덜란드의 귀도 반 로섬(*Guido van Rossum*)이 개발한 대화형 프로그래밍 언어
 - *Python* : 그리스 신화 파르나서스 산의 동굴에 숨어 살다가 아폴로에 의해 살해된 거대한 뱀
 - 만인을 위한 프로그래밍 (*CP4E : Computer Programming for Everybody*) 언어로써 개발됨
 - 배우기 쉽고 사용하기 쉬운 언어
 - 인터프리트 방식 : 실행결과를 즉시 볼 수 있음 (대화형)
 - 무료
 - 생산성이 높고 다른 언어로 작성된 코드와 결합하는 능력이 탁월
 - 다양하고 풍부한 라이브러리
 - 애니메이션이나 그래픽을 쉽게 사용

파이썬의 라이브러리



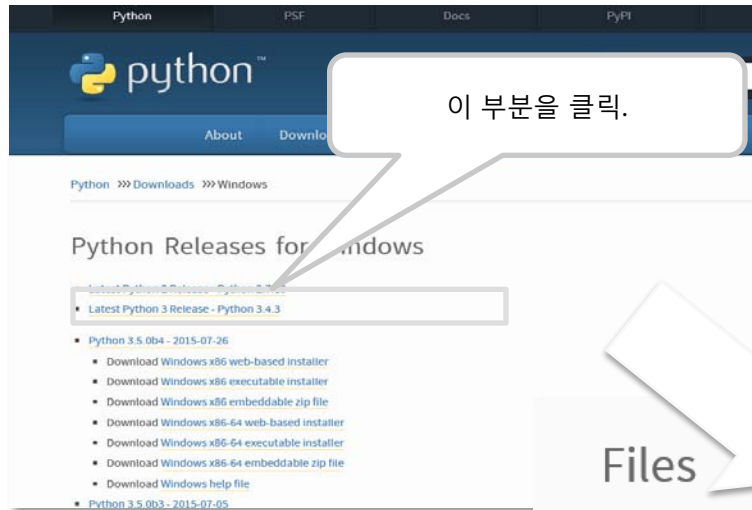
파이썬 설치(1)

❖ 운영체제 확인



파이썬 설치(2)

❖ 다운로드 링크: <https://www.python.org/downloads/windows/>



Files

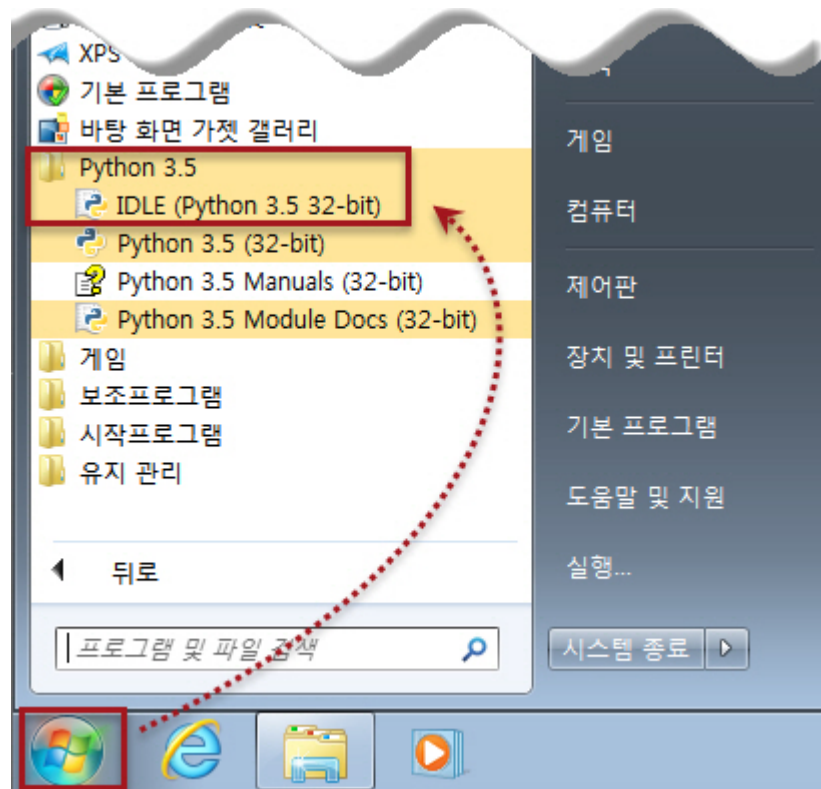
Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later
Windows debug information files	Windows	
Windows debug information files for 64-bit binaries	Windows	
Windows help file		
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64, not Itanium processors
Windows x86 MSI installer		

64비트 윈도우용 설치 파일

32비트 윈도우용 설치 파일

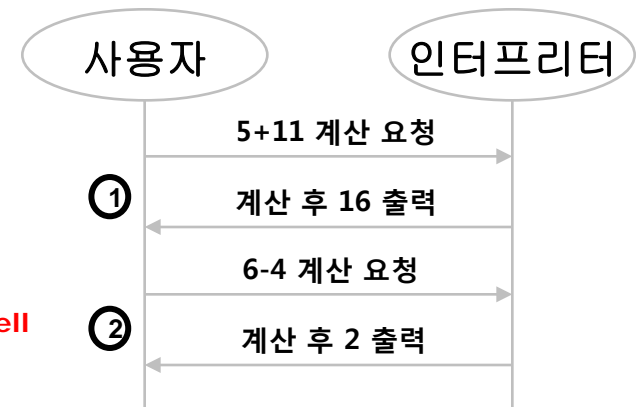
파이썬 실행(1)

- ❖ 윈도우의 시작 메뉴에서 "IDLE" 프로그램을 찾아서 실행



파이썬 실행(2)

- ❖ *Python Shell*은 파이썬의 통합 개발 환경 (*IDLE: Integrated Development Environment*)
 - 개발을 효율적으로 수행하기 위해서 제공하는 대화형 실행도구
 - *IDLE*: 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 소프트웨어
- ❖ 인터프리터 (*Interpreter*)는 *Shell Prompt*를 통해 사용자와 대화함
 - 문장 단위로 사용자의 입력을 *Interpreter*가 처리함



Python Shell의 두 가지 사용방식

❖ 대화형 인터프리터 모드

- 짧고 간단한 예제 코드는 대화형 인터프리터 모드를 이용

❖ 코드 편집기 모드

- 코드의 양이 많고 복잡해지면 인터프리터 모드는 번거로움
- 또한 추후에 프로그램을 재사용하기 위해서 파일로 저장한 후 사용

이걸 한 줄씩 입력하라고?



```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s" % (nodename, label)
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s];' % ast[1]
        else:
            print ']'
    else:
        print '["';
        children = []
        for n, childenumerate(ast[1:]):
            children.append(dotwrite(child))
        print ', ' % ast[1] -> {' % nodename
        for n, namechildren
            print '%s' % name,
```


코드 편집기 모드

- ❖ 텍스트 에디터를 이용하여 명령어들을 파일에 저장한 후에 파일을 읽어서 명령어들을 하나씩 실행하는 방식
 - 명령어들이 저장된 파일을 소스파일 (*source file*)이라고 함
- ❖ 소스파일 작성
 - 파이썬 쉘의 메뉴에서 [File] -> [New File]
- ❖ 소스파일 저장
 - 텍스트 에디터의 [File]->[Save]
- ❖ 소스파일 실행
 - 파이썬 쉘의 메뉴에서 [Run]->[Run Module]

구문오류 (*Syntax Error*)

```
>>> pront("Hello World")  
SyntaxError: invalid syntax
```

```
>>> 1 +  
SyntaxError: invalid syntax
```

```
>>> 3 +* 2  
SyntaxError: invalid syntax
```

실행오류 (*Run-time Error*)

```
print("안녕하세요? 파이썬에 오신 것을 환영합니다!")  
print("프로그래밍 공부를 즐기셨으면 합니다.")  
print("안녕!" + 3)
```

*print("안녕!" * 3)로 수정해야함*

===== RESTART: D:\s.py =====

안녕하세요? 파이썬에 오신 것을 환영합니다!
프로그래밍 공부를 즐기셨으면 합니다.

Traceback (most recent call last):

File "D:\s.py", line 3, in <module>

print("안녕!" + 3)

TypeError: Can't convert 'int' object to str implicitly

주석 (Comment)

❖ 코드의 이해를 위해 코드 사이에 삽입하는 설명글

- 기호 **#**뒤에 삽입하며 프로그램의 실행 결과에 영향을 끼치지 않음
 - 대화형 인터프리터 모드와 코드 편집기 모드에서 모두 사용가능
 - 예) 주석이 없다면 가격에 왜 1.1을 곱했는지 코드만 보서는 알 수 없음

```
totalPrice = productPrice * 1.1 #10%의 세금을 추가함
```

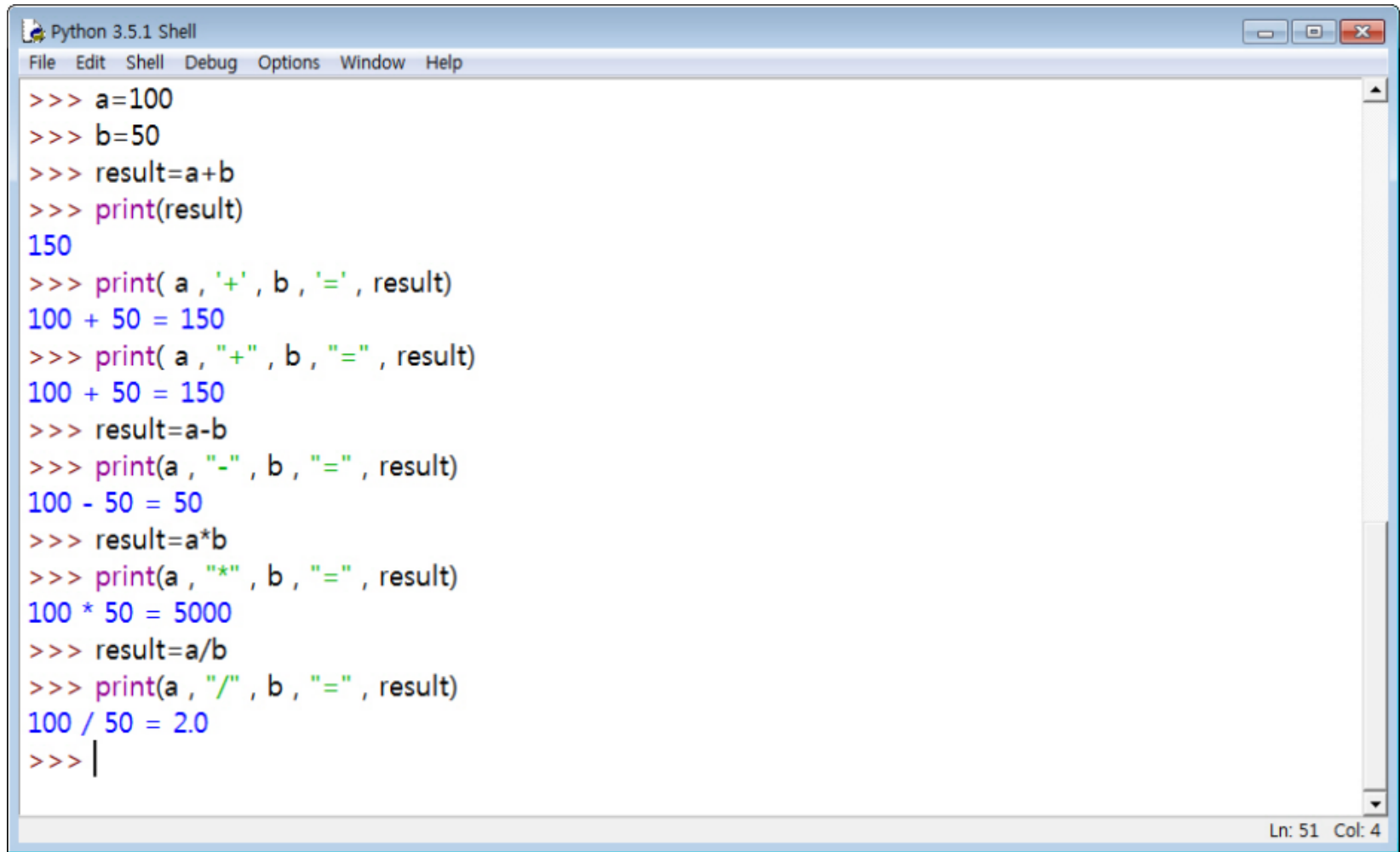
- 따옴표 3개를 이용하여 여러 줄의 주석도 삽입가능
 - 코드 편집기 모드에서만 사용가능하며, 코드가 있는 같은 줄에는 사용불가
 - 큰 따옴표 또는 작은 따옴표 모두 사용가능 (열고 닫을때 일관성있게 사용해야함)

“ ”

모든 제품에는 10%의 세금이 추가되어야 합니다.
부가가치세와 관련된 국내 법규를 지켜야 합니다.
해당 내용은 2015.10.11 부터 적용됩니다. “ ”

```
totalPrice = productPrice * 1.1
```

실습: 계산 프로그램 (1)

A screenshot of a Python 3.5.1 Shell window. The window has 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 perform addition, subtraction, multiplication, and division on the variables 'a' and 'b'. The status bar at the bottom right indicates 'Ln: 51 Col: 4'.

```
>>> a=100
>>> b=50
>>> result=a+b
>>> print(result)
150
>>> print( a , '+' , b , '=' , result)
100 + 50 = 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
>>> |
```

실습: 계산 프로그램 (2)

- = 기호는 ‘같다’ 는 의미가 아니라 ‘오른쪽의 것을 왼쪽으로 넣는다’ 는 의미
 - $a \leftarrow 100$ 과 같은 개념
- 변수(그릇)의 개념

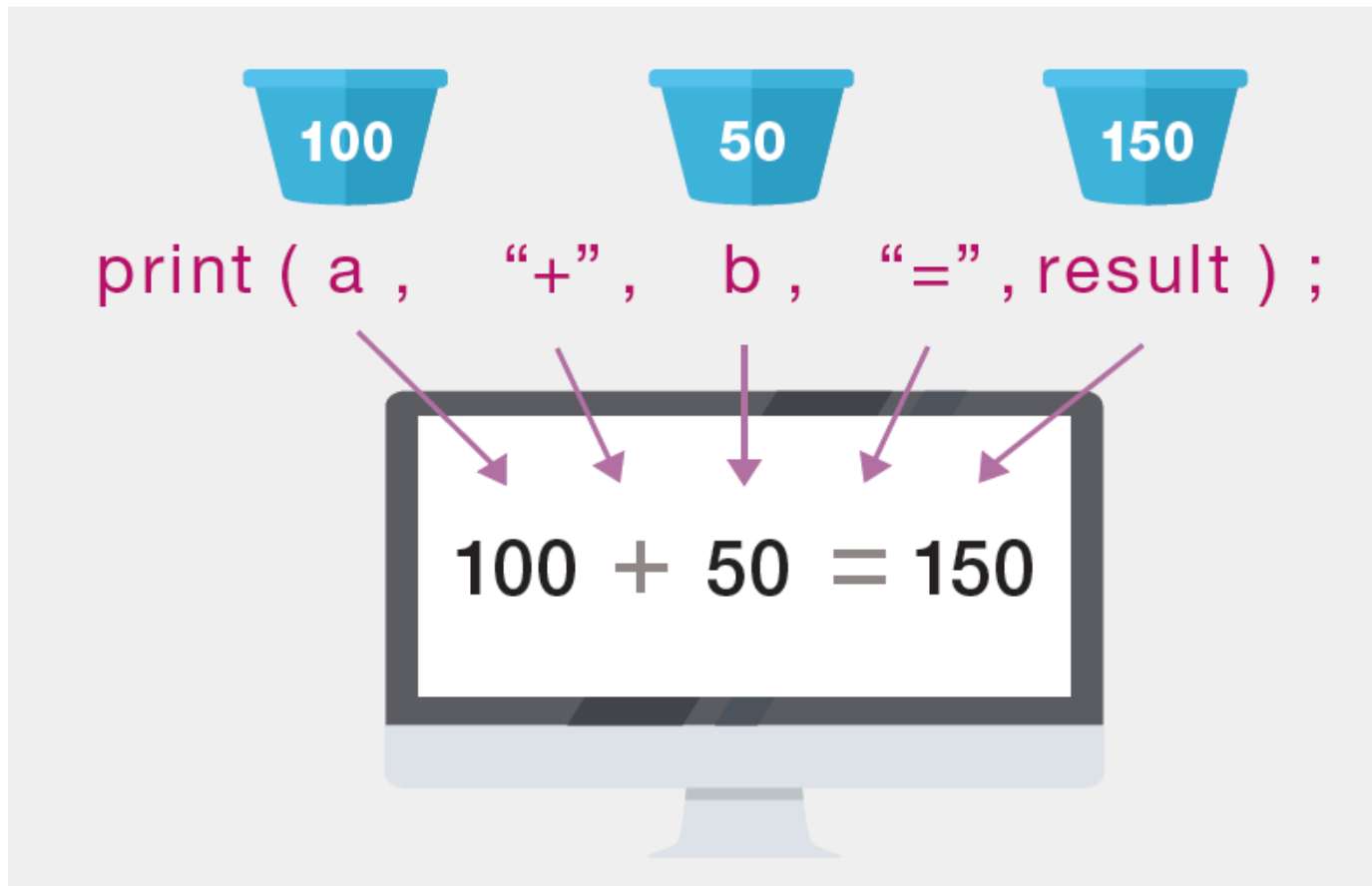


- $result = a + b$



실습: 계산 프로그램 (3)

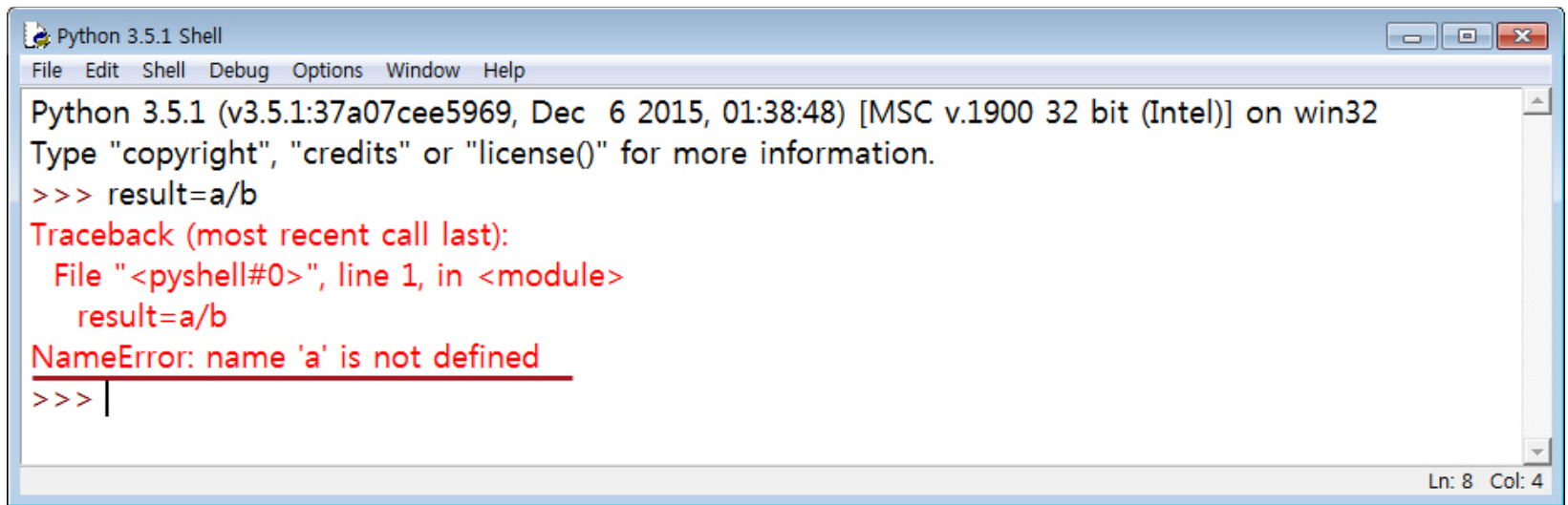
- `print()`를 사용한 출력



실습: 프로그램 저장 (1)

❖ 사용했던 변수의 재사용

- 프로그램을 종료 후 **IDLE**을 다시 실행하여 앞에서 실습한 결과를 실행 시에 에러가 남
- 프로그램 종료 시에 변수는 메모리에 저장되지 않고 사라짐



The screenshot shows a Python 3.5.1 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The main text area displays the following content:

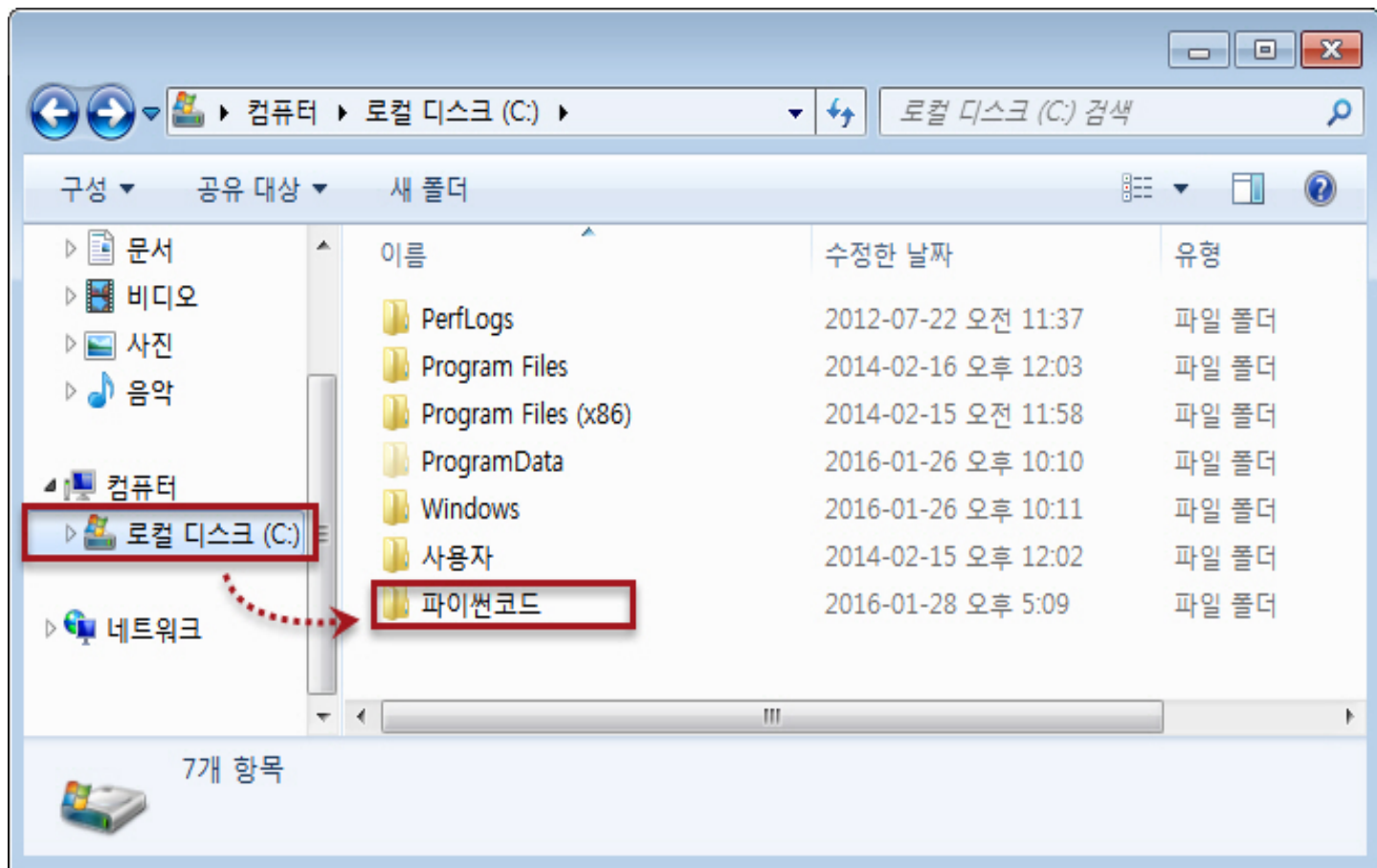
```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> result=a/b
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    result=a/b
NameError: name 'a' is not defined
>>> |
```

The status bar at the bottom right indicates "Ln: 8 Col: 4".

- 따라서 추후에 프로그램을 재사용하려면 하드디스크에 파일로 저장해야 함

실습: 프로그램 저장 (2)

❖ 실습 폴더 생성의 예



실습: 프로그램 저장 (3)

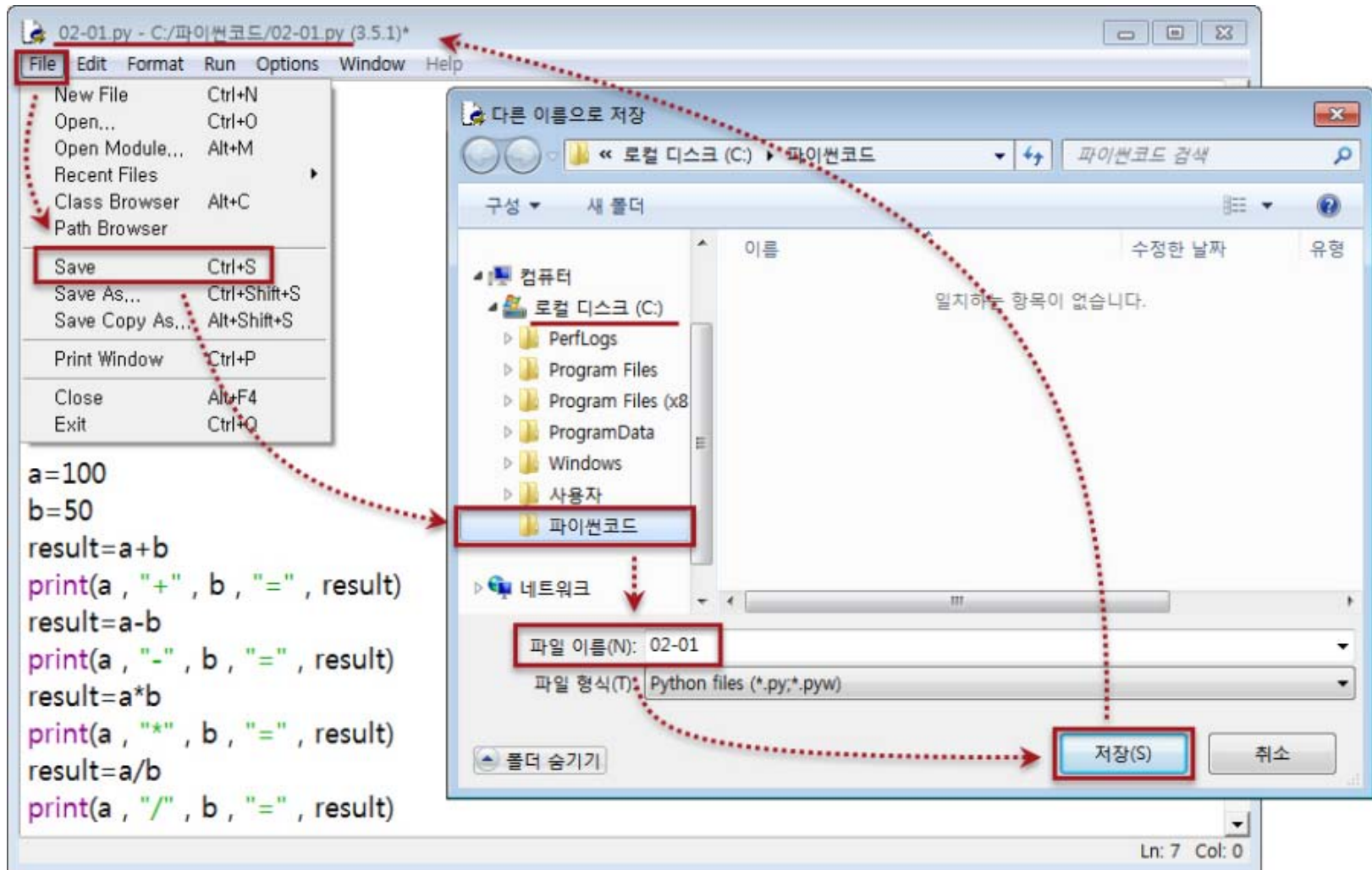
❖ 파일 생성

■ IDLE의 [File] [New File] 메뉴

```
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)
```

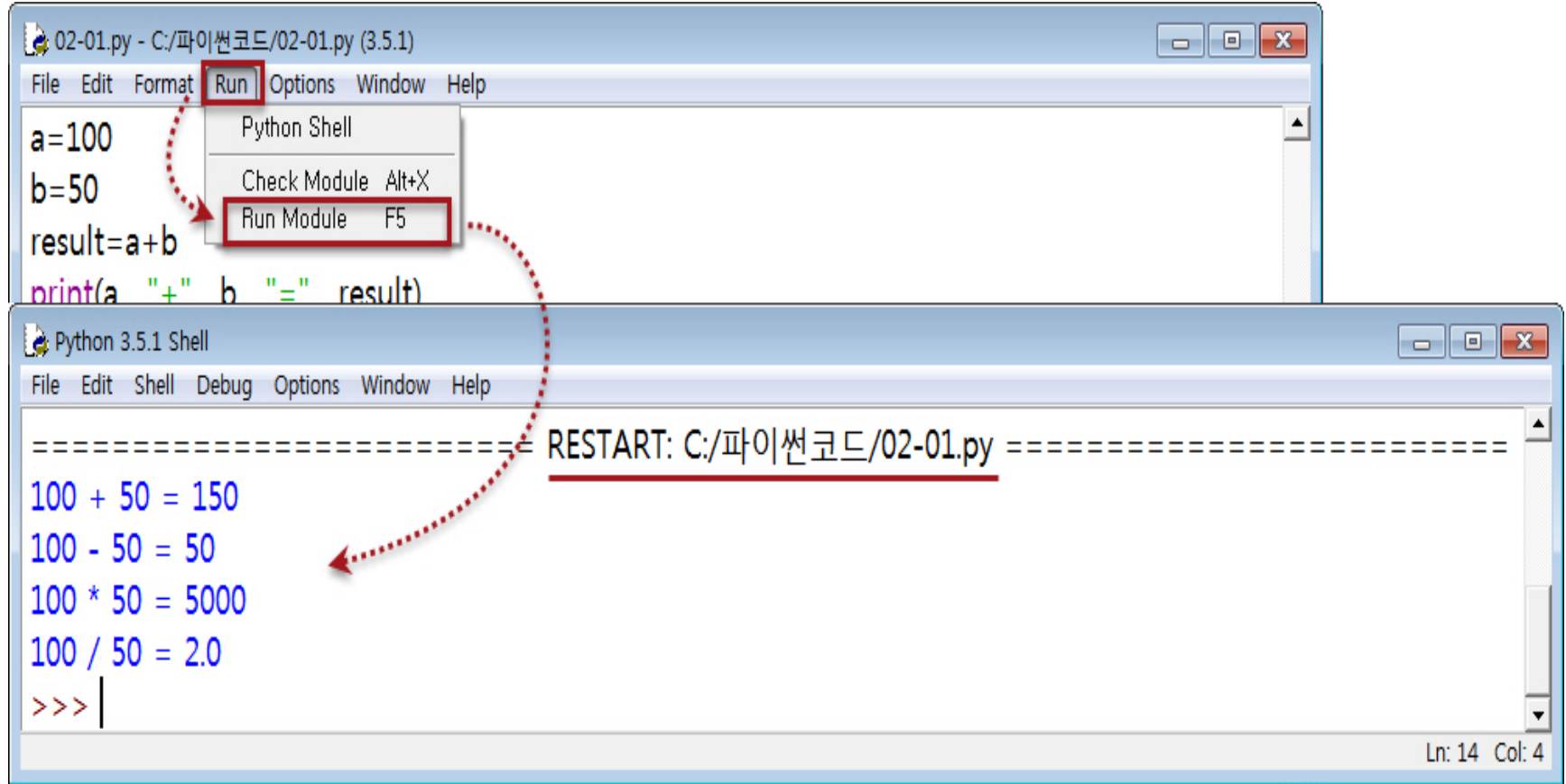
실습: 프로그램 저장 (4)

❖ 파일 저장 (cf: 확장자 ~.py)



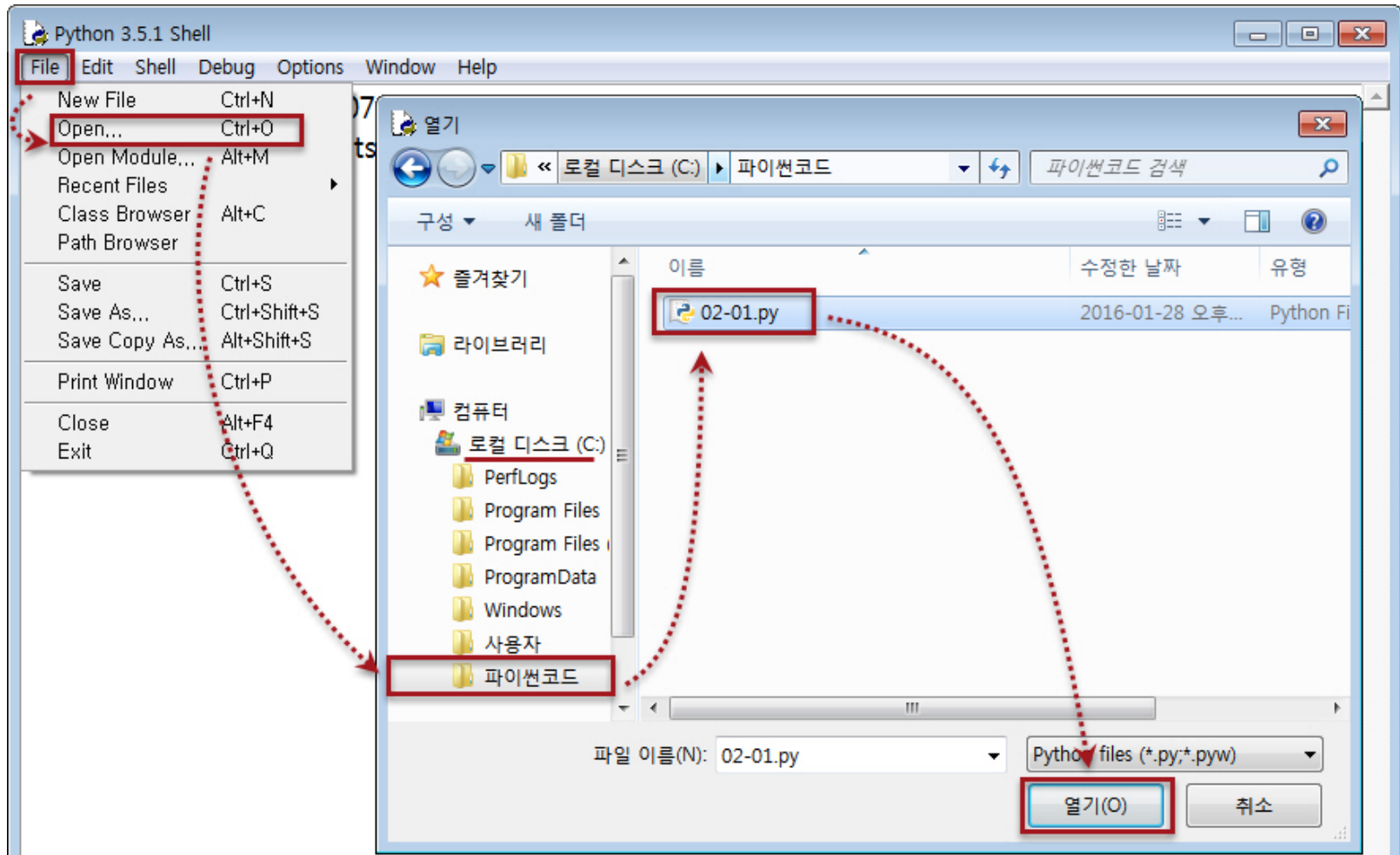
실습: 프로그램 저장 (5)

❖ 파일 실행



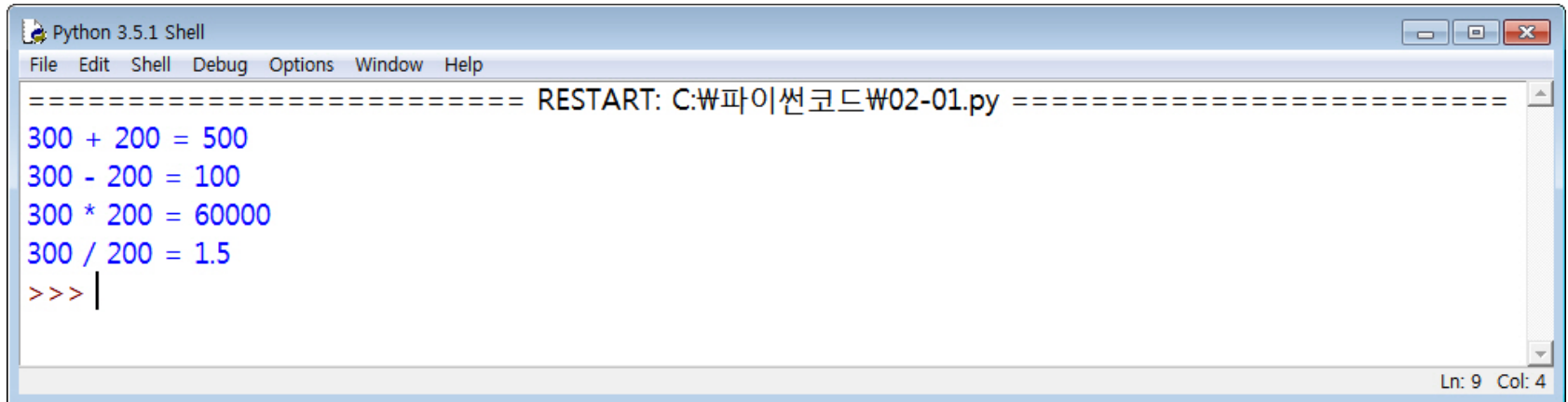
실습: 프로그램 저장 (6)

❖ 파일 열기



실습: 프로그램 저장 (7)

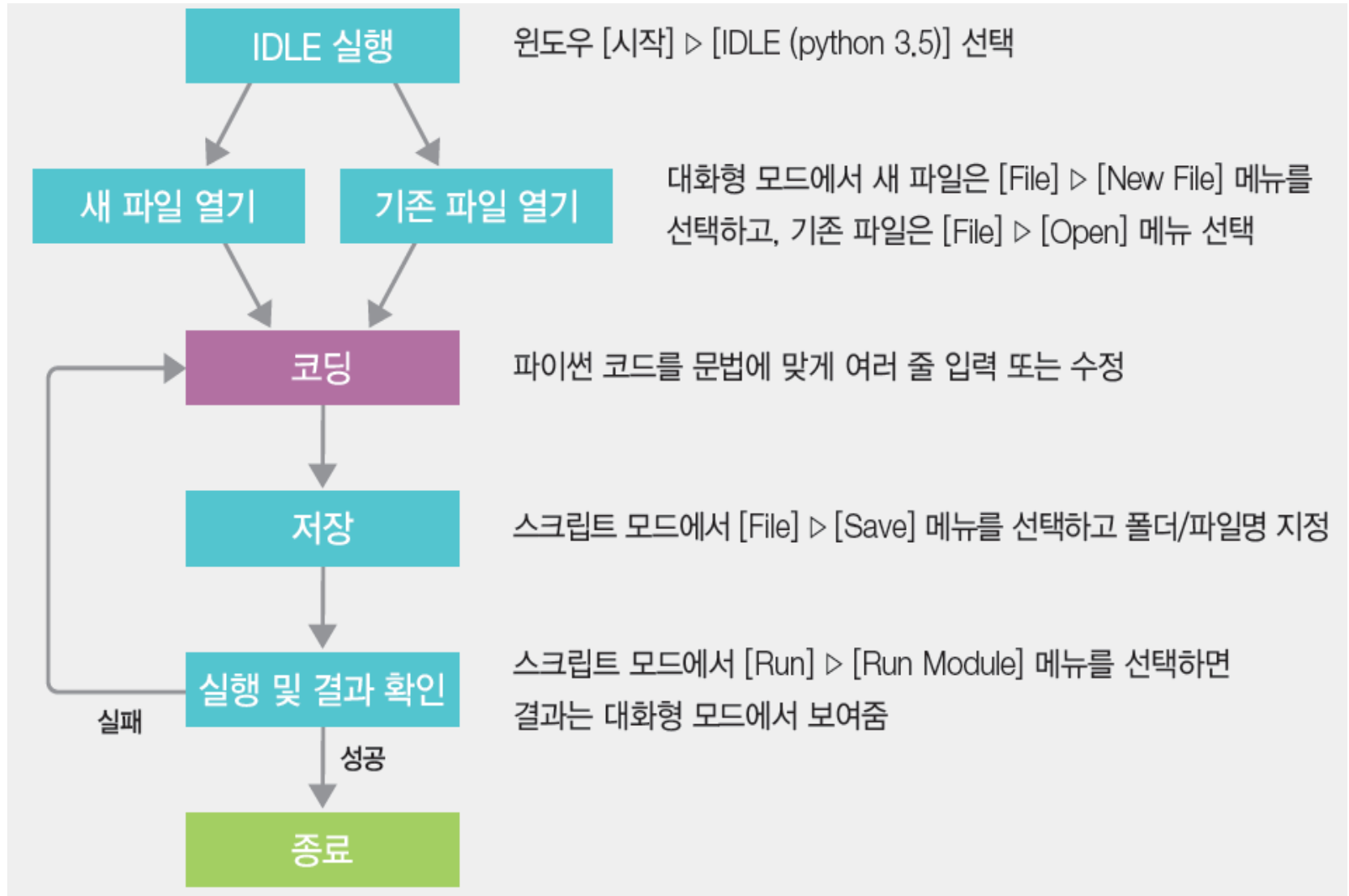
❖ *a와 b 값을 300과 200으로 고친 후 실행*



A screenshot of a Python 3.5.1 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the output of a program restart: '==== RESTART: C:\W파이썬코드\W02-01.py ====='. Below this, four arithmetic operations are displayed in blue text: '300 + 200 = 500', '300 - 200 = 100', '300 * 200 = 60000', and '300 / 200 = 1.5'. At the bottom left of the text area, the prompt '>>>' is followed by a vertical cursor. The status bar at the bottom right indicates 'Ln: 9 Col: 4'.

```
==== RESTART: C:\W파이썬코드\W02-01.py =====
300 + 200 = 500
300 - 200 = 100
300 * 200 = 60000
300 / 200 = 1.5
>>> |
```

요약: 코드 편집기 모드 사용법



실습: 계산 프로그램 업그레이드 (1)

❖ 값을 키보드로 입력받기

a에 100을 넣습니다. → a에 입력할 값을 키보드로 입력받습니다.
b에 50을 넣습니다. → b에 입력할 값을 키보드로 입력받습니다.



실습: 계산 프로그램 업그레이드 (2)

❖ input() 함수로 값 입력받기

```
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)
```

실습: 계산 프로그램 업그레이드 (3)

❖ 실행 후 에러발행

- **input()** 함수는 값을 입력받지만, 모두 문자열로 취급하기 때문에 오류가 발생한 것임

```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/파이썬코드/02-02.py =====
100
50
100 + 50 = 10050
Traceback (most recent call last):
  File "C:/파이썬코드/02-02.py", line 5, in <module>
    result=a-b
TypeError: unsupported operand type(s) for -: 'str' and 'str'
>>>
```

Ln: 18 Col: 4

실습: 계산 프로그램 업그레이드 (4)

❖ int() 함수를 이용해 정수로 변환하기

`int("100")` → 결과는 정수 100

`int(100.123)` → 결과는 정수 100

```
a=int(input())
```

```
b=int(input())
```

