

Section 01 프로그래밍 언어의 개념과 종류

■ 프로그래밍 언어의 개념

- 프로그래밍 언어 : 컴퓨터가 이해하는 말로 컴퓨터에서 작동하는 소프트웨어(엑셀, 한글, 인터넷 익스플로러 등)를 만드는 도구
- 프로그래머 : 프로그래밍 언어를 사용해 소프트웨어나 앱을 만드는 사람



그림 1-1 프로그래머, 프로그래밍 언어, 소프트웨어

Section 01 프로그래밍 언어의 개념과 종류

■ 프로그래밍 언어의 종류

- 수백 가지가 넘는 종류 중 많이 사용되는 프로그래밍 언어는 C/C++, 자바(Java), HTML, PHP, 파이썬 등



그림 1-2 스포츠와 프로그래밍 언어 비교

Section 02 파이썬 소개

■ 파이썬 역사

- 배우기도 쉽고 결과도 바로 확인할 수 있어 초보자에게 적합한 프로그래밍 언어
- 귀도 반 로섬(1956년~)이라는 프로그래머가 C 언어로 제작해 1991년에 공식으로 발표
- 사전적인 의미는 비단뱀으로 로고도 파란색과 노란색 비단뱀 두 마리가 서로 얹혀 있는 형태



그림 1-3 파이썬 로고(출처 : <https://www.python.org>)



그림 1-4 파이썬의 창시자 귀도 반 로섬(출처 : 위키피디아)

Section 02 파이썬 소개

■ 파이썬 특징

- ❶ 강력한 기능을 무료로 사용할 수 있다
 - 파이썬은 오픈 소스이며, 비용을 지불하지 않고 무료로 사용 가능. 다양한 추가 라이브러리도 무료
- ❷ 읽기 쉽고 사용하기 쉽다
 - 직관적인 코드를 사용해 C나 자바 같은 언어보다 읽기 쉬운 프로그램을 빨리 제작할 수 있어 비용 절감 효과 제공
- ❸ 사물인터넷과 잘 연동된다
 - 라즈베리파이 기반의 사물인터넷이 파이썬을 잘 지원하므로 사물인터넷 개발 및 운영에 적극 활용
- ❹ 다양하고 강력한 외부 라이브러리들이 풍부하다
 - 파이썬에서 제공하는 라이브러리뿐 아니라, 외부에서 제공하는 다양한 서드 파티(Third Party) 라이브러리까지 사용 가능
- ❺ 강력한 웹 프레임워크를 사용할 수 있다
 - 파이썬의 웹 프레임워크를 사용해 강력하고 빠른 웹 환경을 구축 가능

Section 02 파이썬 소개

- 파이썬의 단점
 - 느린 속도
 - 파이썬은 컴파일러 언어가 아닌 스크립트 언어이기 때문에 컴파일러 언어보다 느림
→ 이를 보완하려고 많은 파이썬 패키지를 최적화시키고 있음
 - 모바일 컴퓨팅 분야에 지원이 약하고 하드웨어 제어 등과 관련된 부분 사용이 어려움
- 파이썬의 실행 화면
 - print("Hello, world!")를 입력한 후 [Enter]를 눌러 Hello, world!를 출력한 화면
 - print는 무언가를 프린트하라는 의미이므로 print()에서 괄호 안에 있는 것을 화면에 출력

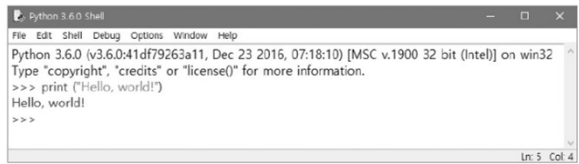


그림 1-5 파이썬 실행 화면

Section 02 파이썬 소개

여기서 잠깐

컴파일러(Compiler) 언어는 소스 코드를 실행 가능한 기계어로 일괄 번역한 후 번역이 완료된 파일(*.exe, *.class 등의 파일)을 실행하는 언어를 의미한다. 소스 코드를 기계어로 번역하는 과정을 컴파일(Compile)이라고 하며, 이 작업을 하는 프로그램을 컴파일러라고 한다. 대표적인 컴파일러 언어로는 C/C++, 자바 등이 있다. 이와 달리 스크립트 언어 또는 인터프리터 언어는 소스 코드를 한 줄씩 읽어 실행되어 별도의 실행 파일이 생성되지 않는데, 이때 한 줄씩 처리하는 프로그램을 인터프리터(Interpreter)라고 한다. 대표적인 스크립트 언어로는 파이썬, 자바스크립트(JavaScript), 펄(Perl) 등이 있다.

일반적으로 컴파일러 언어는 소스 코드를 일괄적으로 기계어로 번역해 놓기 때문에 실행 속도가 스크립트 언어보다 빠르다. 하지만 컴파일러 언어는 배우는 데 시간이 오래 걸리는 반면, 스크립트 언어는 대부분 빠른 시간에 배울 수 있다는 장점이 있다.

컴파일러 언어와 스크립트 언어

Section 02 파이썬 소개

여기서 잠깐

Hello World 프로그램 다른 프로그래밍 언어를 공부한 적이 있다면, 화면에 Hello, world를 출력하는 Hello World 프로그램에 익숙할 것이다. 이 프로그램은 대부분의 프로그래밍 언어 책에서 처음으로 만드는 기본 예제이다. [그림 1-5]도 이런 맥락에서 작성한 것이다. 다음은 C나 자바로 작성된 Hello World 프로그램이다.

C 프로그램	자바 프로그램
<pre>#include <stdio.h> int main() { printf("Hello, world!\n"); return 0; }</pre>	<pre>public class HelloWorldApp { public static void main(String[] args) { System.out.println("Hello, world!"); } }</pre>

Hello World 프로그램은 1978년에 출판된 브라이언 커니핸과 데니스 리치가 쓴 『The C Programming Language』에서 처음 사용된 것으로 알려져 있다.

Section 03 파이썬 설치와 실행

- 파이썬 다운로드 전 준비 과정(윈도의 확장명 표시)
 - 윈도우 10은 파일 탐색기 실행→[보기] 메뉴 선택 → '파일 확장명' 체크
 - 윈도우 7은 [구성]-[폴더 및 검색 옵션] 메뉴 선택 → [폴더 옵션] 대화상자의 [보기] 탭 클릭 → '알려진 파일 형식의 파일 확장명 숨기기'의 체크 표시 해제 → [확인] 버튼



그림 1-6 윈도우 7 및 윈도우 10에서 확장명 표시

Section 03 파이썬 설치와 실행

- 파이썬 다운로드
 - <http://www.python.org/>에 접속 → [Downloads]-[Download Python 3.x.x] 클릭
 - 설치 파일인 python-3.x.x.exe를 원하는 위치에 저장

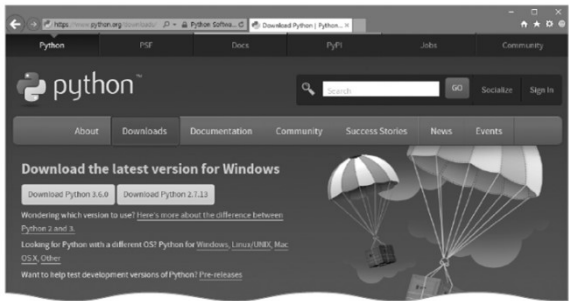


그림 1-7 파이썬 다운로드 화면

Section 03 파이썬 설치와 실행

여기서 잠깐

파이썬 2.x와 3.x

파이썬은 크게 2.x 버전과 3.x 버전으로 나누는데, 이 책을 집필하는 시점에는 각각 2.7.13 버전과 3.6.0 버전까지 발표되었다. 이렇게 나누는 가장 큰 이유는 두 버전이 호환되지 않아 2.x로 작성된 코드를 3.x에서 사용하려면 코드를 일부 수정해야 하기 때문이다.

기존에 작성된 파이썬 코드들은 2.x로 된 것이 많다. 이런 코드들을 일괄적으로 3.x로 수정하기도 어렵고, 잘 작동하는 코드를 굳이 변경할 필요도 느끼지 못하기 때문이다. 하지만 파이썬 2.x가 2.7 버전을 마지막으로 더 이상 업데이트되지 않아 향후에는 3.x 버전만 업데이트될 것이다. 2.x에서 작성한 코드를 사용할 경우가 아니라면 가능한 3.x 버전을 사용하는 것이 최신의 파이썬 기능을 100% 활용할 수 있는 방법이다.

이 책에서도 집필 시점을 기준으로 최신인 3.6.0을 사용할 것이다. 이후에 더 버전을 업데이트해도 별 문제가 없을 것이라 생각되지만, 책과 완전히 동일한 버전을 사용하고 싶다면 이 책의 자료실 (<http://www.hanbit.co.kr/src/4359>)에서 python-3.6.0.exe 파일을 다운로드해서 사용한다.

Section 03 파이썬 설치와 실행

- 파이썬 설치
 - python-3.x.x.exe를 더블클릭 실행 → Add Python 3.6 to PATH에 체크
 - <Install Now> 버튼 클릭(<Customize Installation> 버튼으로 설치 폴더 변경 가능)
 - 설치 진행 → 설치를 마치면 <Close> 버튼 클릭

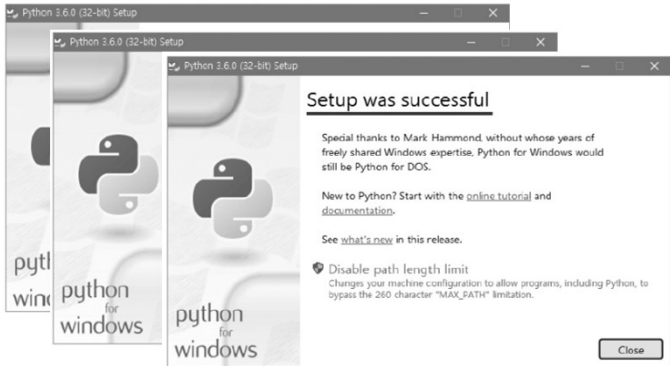


그림 1-8 파이썬 설치 화면

Section 03 파이썬 설치와 실행

- 파이썬 실행
 - 원도의 <시작> 버튼
 - [모든 프로그램]-[Python 3.6]-[IDLE (Python 3.6 32-bit)] 메뉴 선택

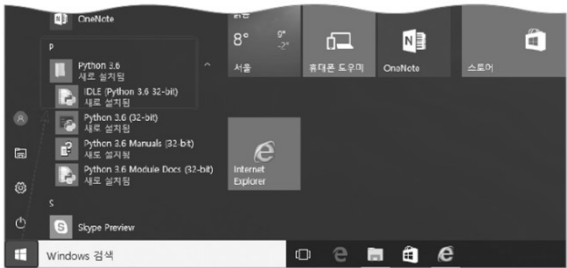


그림 1-9 IDLE 실행

Section 03 파이썬 설치와 실행

- 파이썬 실행
 - 윈도우의 <시작> 버튼
 - [모든 프로그램]-[Python 3.6]-[IDLE (Python 3.6 32-bit)] 메뉴 선택
 - IDLE이 시작되며 파이썬 셸(Python Shell)이 대화형 모드로 나타남

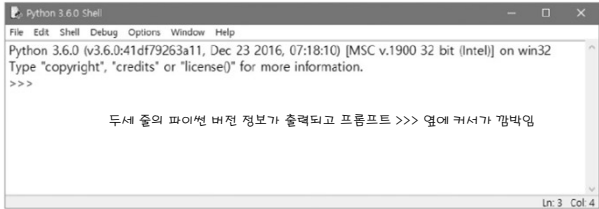


그림 1-10 IDLE 실행 화면

Section 03 파이썬 설치와 실행

- 파이썬 코드 입력과 실행 예
 - 예1
 - >>> 다음에 print("Hello, world!")를 입력하고 [Enter]

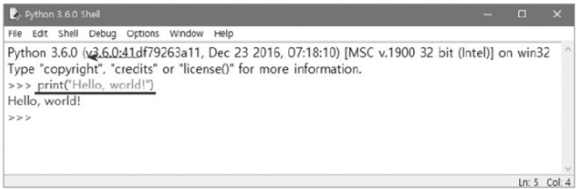


그림 1-11 코드 입력과 실행 예 1

Section 03 파이썬 설치와 실행

- 파이썬 코드 입력과 실행 예
 - 예2
 - >>> 다음에 다음 계산식을 입력하고 [Enter]



그림 1-12 코드 입력과 실행 예 2

Section 03 파이썬 설치와 실행

- 파이썬 코드 입력과 실행 예
 - 예3
 - >>> 다음에 다음 계산식을 입력하고 [Enter]



그림 1-13 코드 입력과 실행 예 3

Section 03 파이썬 설치와 실행

- 파이썬 IDLE 종료
 - [File]-[Exit] 메뉴 선택

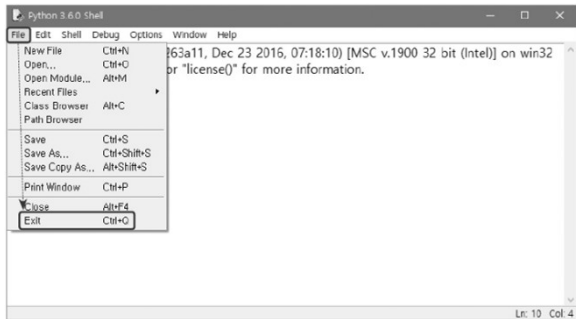
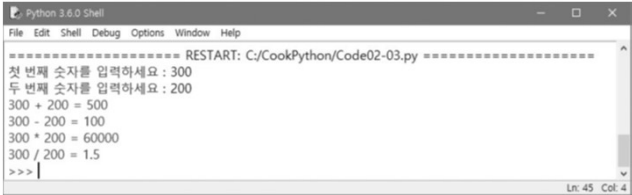


그림 1-14 IDLE 종료

Section04 예제 프로그램

- [프로그램] 간단 계산기
 - 숫자를 2개 입력해 더하기, 빼기, 곱하기, 나누기 등을 계산하는 아주 기본적인 기능



Section04 예제 프로그램

- 프로그램을 작성하는 순서

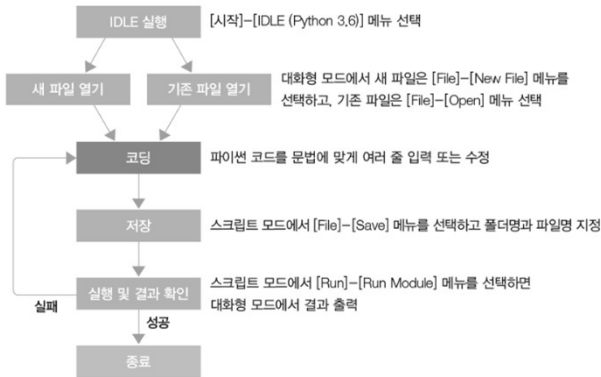


그림 2-18 긴 프로그램을 코딩하는 순서

Section04 계산기 프로그램의 기본 기능 구현

- 필요한 변수 준비
 - = 기호는 같다는 의미가 아니라 '오른쪽의 것을 왼쪽으로 넣어라'는 의미의 대입 연산자
 - a=100은 a ← 100과 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

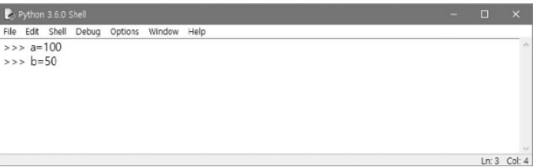


그림 2-1 변수 준비



Section04 계산기 프로그램의 기본 기능 구현

필요한 변수 준비

메모리 (memory)	주소 (address)	변수 (variable)
	0x0007	
	0x0006	
	0x0005	
	0x0004	
7	0x0003	b
3	0x0002	a
Sungchul Choi	0x0001	professor

[메모리의 주소 할당]

Section04 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

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
>>> a=100
>>> b=50
>>> result=a+b
```

그림 2-3 더하기 구현



그림 2-4 더하는 작업

Section04 계산기 프로그램의 기본 기능 구현

더한 결과 출력

- result 그릇의 내용만 출력

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
>>> a=100
>>> b=50
>>> result=a+b
>>> print(result)
150
```

그림 2-5 더한 결과 출력 1

- result 그릇의 내용과 계산식도 출력

```
Python 3.6.0 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
```

그림 2-6 더한 결과 출력 2

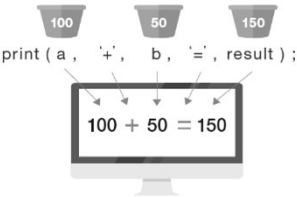


그림 2-7 print() 함수로 모두 출력

Section04 계산기 프로그램의 기본 기능 구현

여기서 잠깐

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.

Section05 계산기 프로그램 저장

■ 빼기, 곱하기, 나누기 기능 구현

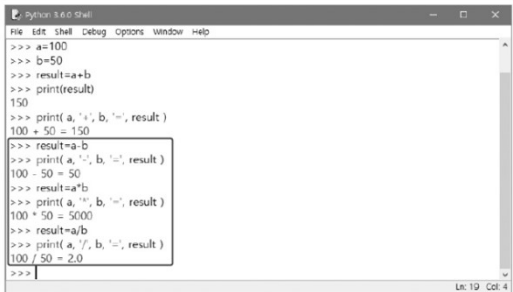


그림 2-8 실행 최종 결과

Section05 계산기 프로그램 저장

■ IDLE 종료

- exit() 코드를 입력한 후 Kill? 메시지창에서 <확인> 버튼 클릭(또는 [File]-[Exit] 메뉴 선택)



그림 2-9 IDLE 종료

Section05 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

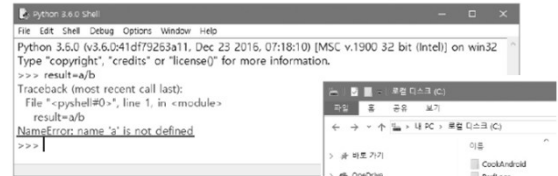


그림 2-10 오류 발생

- 이 책은 C 드라이브(C:\) 바로 아래에 CookPython 폴더를 만들어 사용

그림 2-11 실행용 폴더 생성

Section05 계산기 프로그램 저장

■ 파이썬 파일 저장(1)

- 코드가 수식 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

Code02-01.py

```
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)
```

Section05 계산기 프로그램 저장

- 파이썬 파일 저장(2)
- 스크립트 모드에서 [File]-[Save] 메뉴를 선택해 C:\CookPython\ 폴더에 Code02-01 이름으로 저장(확장명 *.py가 자동으로 붙음)



그림 2-12 파이썬 코드를 파일로 저장

Section05 계산기 프로그램 저장

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



그림 2-13 파이썬 코드를 파일로 실행

- 파이썬 코드를 파일 탐색기에서 바로 실행하는 방법은 교재 41쪽의 [여기서 잠깐] 참고

Section05 계산기 프로그램 저장

- 파이썬 파일 열기(1)
- IDLE에서 [File]-[Open] 메뉴를 선택한 후 Code02-01.py 열기

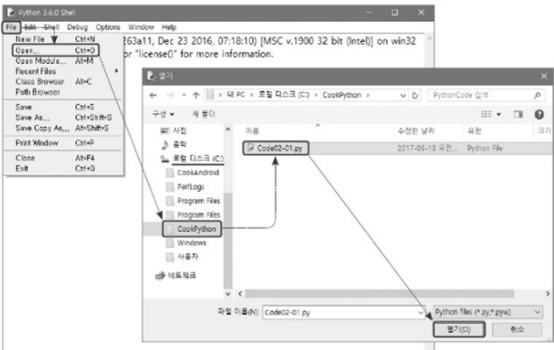


그림 2-16 파이썬 파일 열기

Section05 계산기 프로그램 저장

- 파이썬 파일 열기(2)
- IDLE에서 [File]-[Open] 메뉴를 선택한 후 Code02-01.py 열어 a와 b의 값을 300과 200으로 수정한 후 [File]-[Save] 메뉴 선택(또는 [Ctrl] + [S])해 저장 후 [Run]-[Run Module] 메뉴(또는 [F5])로 다시 실행

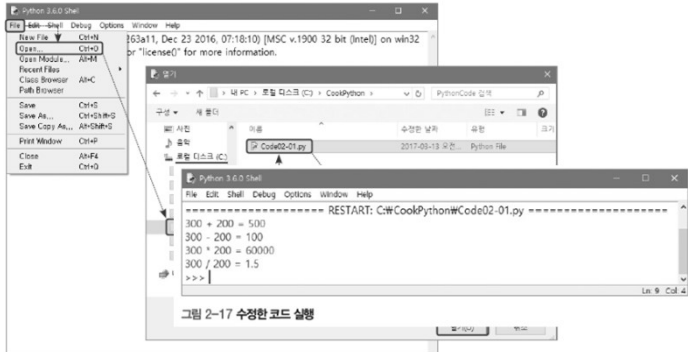


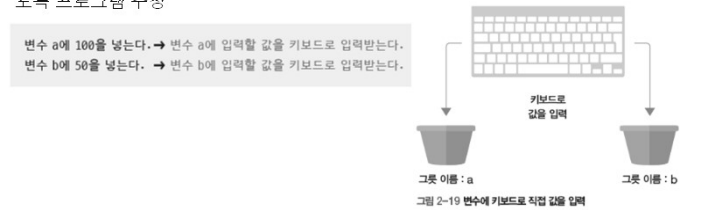
그림 2-17 수정한 코드 실행

그림 2-16 파이썬 파일 열기

Section06 계산기 프로그램 확장

파이썬 파일 만들어 저장

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



- 대화형 모드에서 [File]-[New File] 메뉴를 선택해 새 파일을 연 후 스크립트 모드에서 [File]-[Save] 메뉴를 선택해 C:\CookPython\Code02-02.py로 저장

Section06 계산기 프로그램 확장

input() 함수를 사용해 값 입력

- Code02-01.py의 1~2행을 input() 함수를 사용하도록 수정 → [F5]를 눌러 실행 → 숫자 하나를 입력하고 [Enter] → 다시 숫자 하나를 입력하고 [Enter]

```
Code02-02.py
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)
```

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



Section06 계산기 프로그램 확장

input() 함수를 사용해 정수로 변환

- 오른쪽 예처럼 int() 함수를 사용해 정수로 변환

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

- Code02-02.py의 1~2행을 다음과 같이 수정 후 다시 [F5]를 눌러 실행

```
Code02-03.py
1 a = int(input())
2 b = int(input())
```



Section06 계산기 프로그램 확장

계산기의 최종 버전

```
Code02-04.py
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)
```

