

# 파이썬 프로그래밍 – 02

## 변수

---

컴퓨터소프트웨어공학과  
성낙준



## 이번 장에서 학습하는 내용

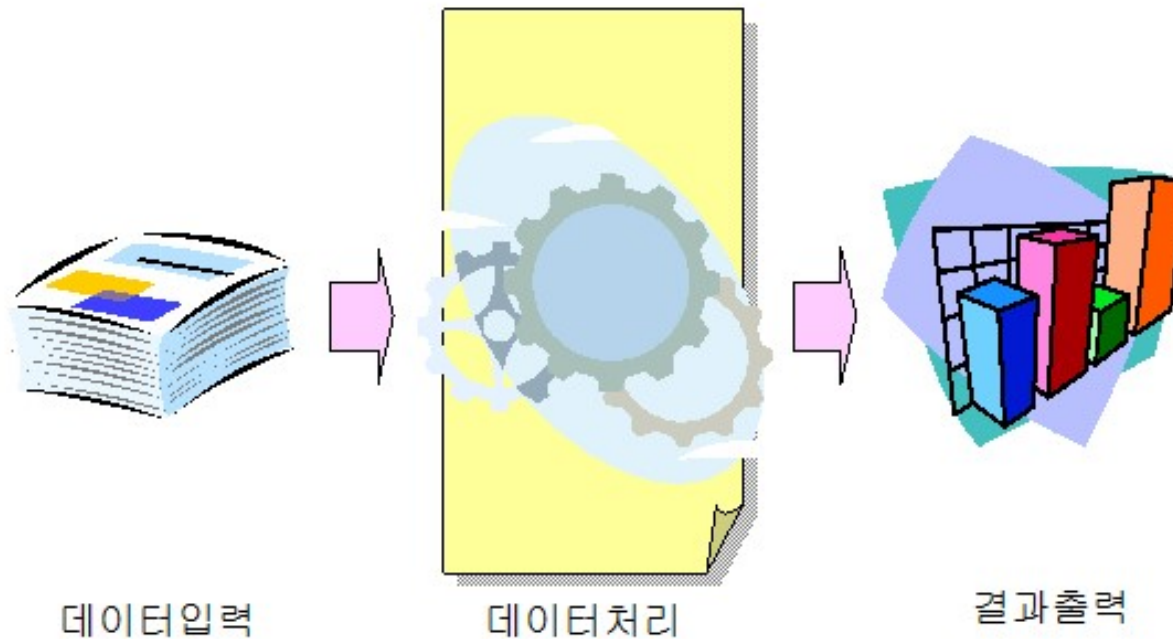
(1) 첫 번째 프로그램은 두 개의 정수를 가지고 덧셈 연산을 실행한 후에, 연산 결과를 화면에 출력한다.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
첫 번째 정수를 입력하시오: 100
두 번째 정수를 입력하시오: 200
100 과 200 의 합은 300 입니다.
>>>
```

Ln: 8 Col: 4



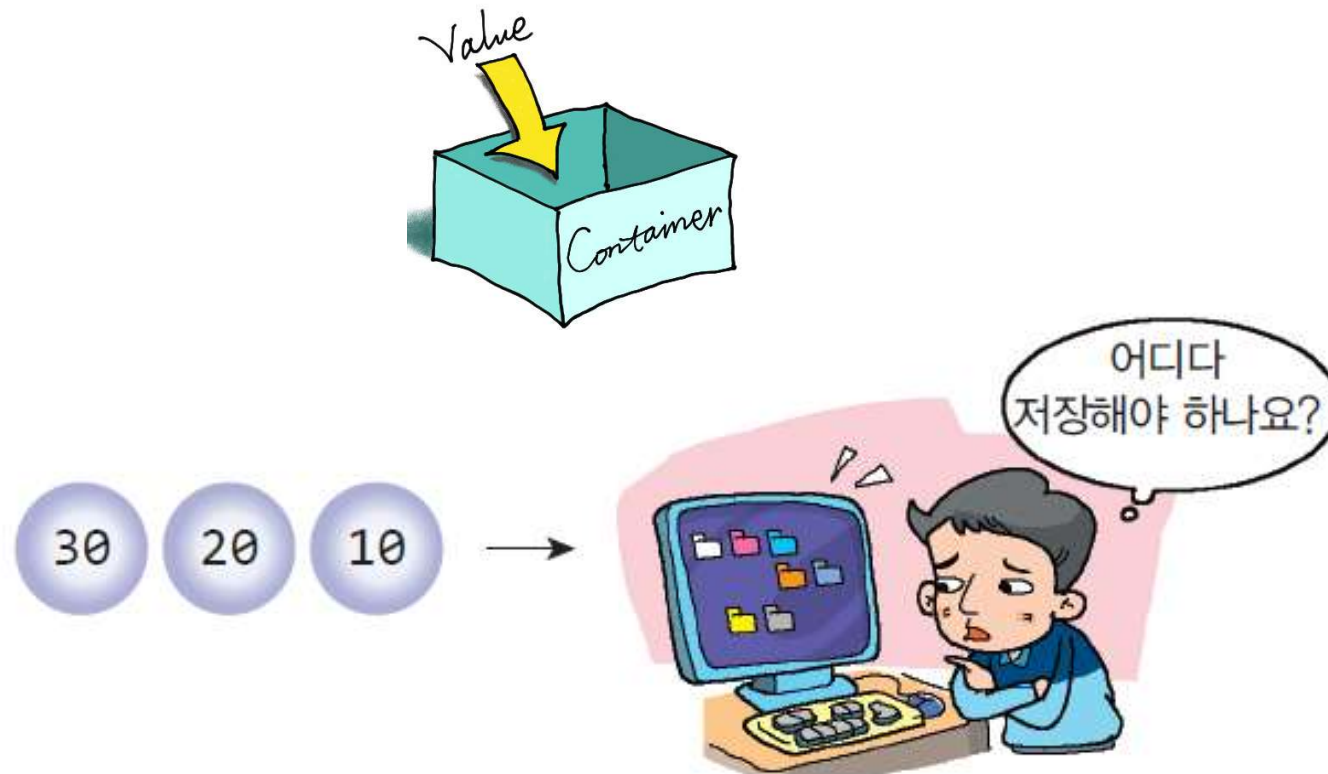
- 일반적인 프로그램은 외부로부터 데이터를 받아서(입력단계), 데이터를 처리한 후에(처리단계), 결과를 화면에 출력(출력단계) 한다.





## ■ 변수

- 변수(variable)는 값을 저장하는 상자로 생각할 수 있다.
- 변수는 컴퓨터 메모리 공간에 만들어 진다.

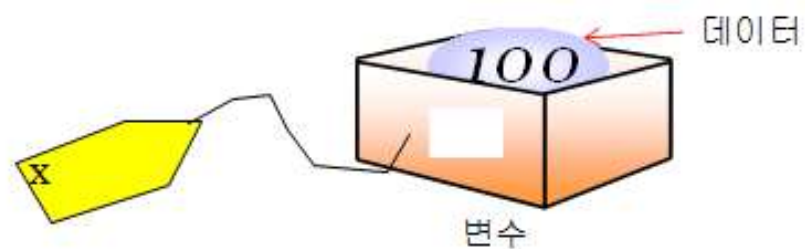




- 파이썬에서 변수를 생성하려면 다음과 같이 한다.

```
>>> x = 100
```

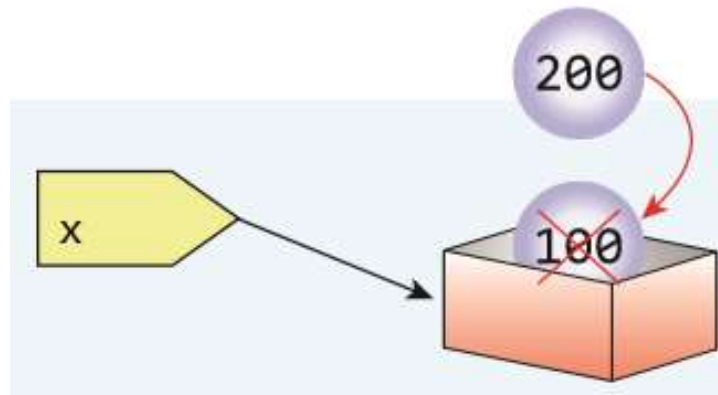
```
>>>
```





- 생성된 변수에는 얼마든지 다른 값을 저장할 수 있다.

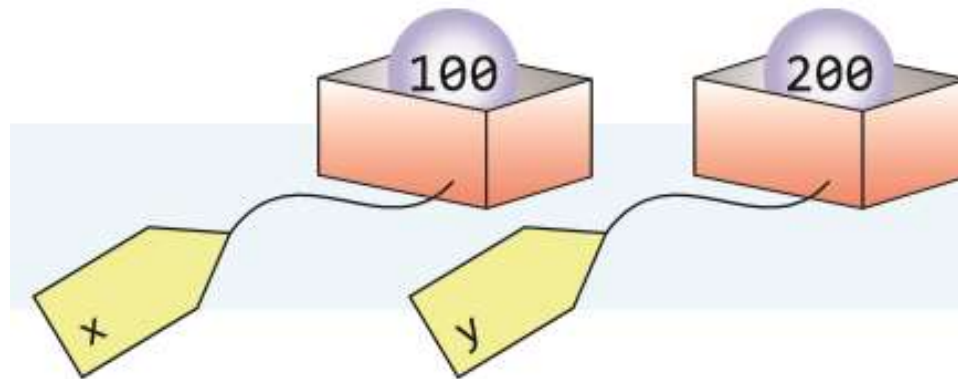
```
>>> x = 100  
>>> x = 200  
>>> print(x)  
200
```





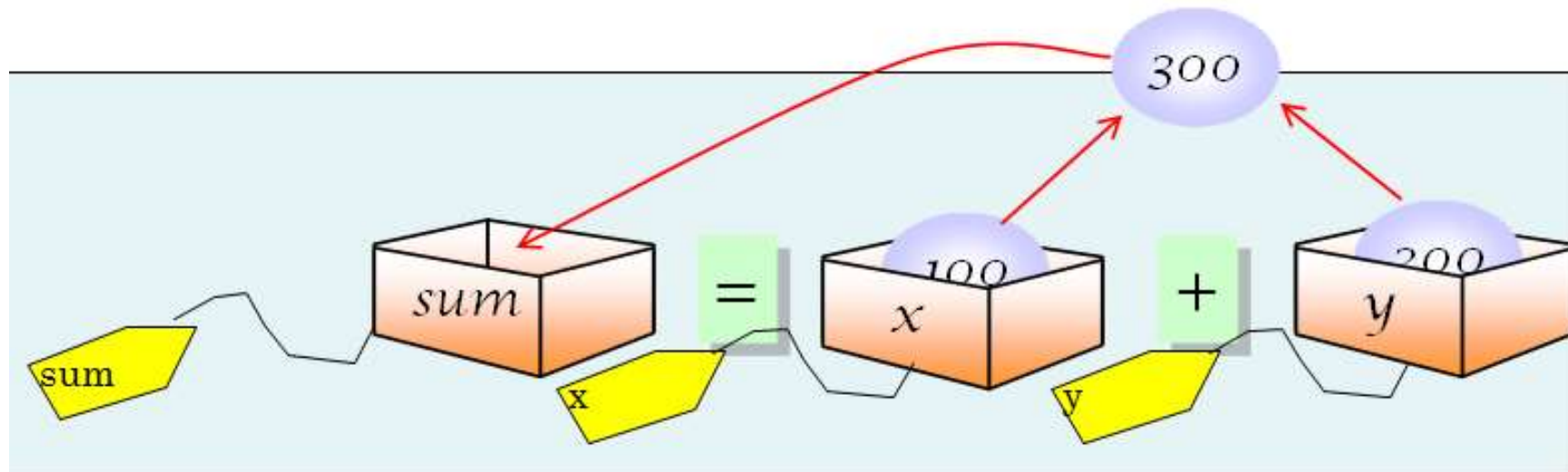
## 변수 2개 생성

```
>>> x = 100  
>>> y = 200
```





```
>>> x = 100  
>>> y = 200  
>>> sum = x + y  
>>> print(sum)  
300
```







## 주의!!

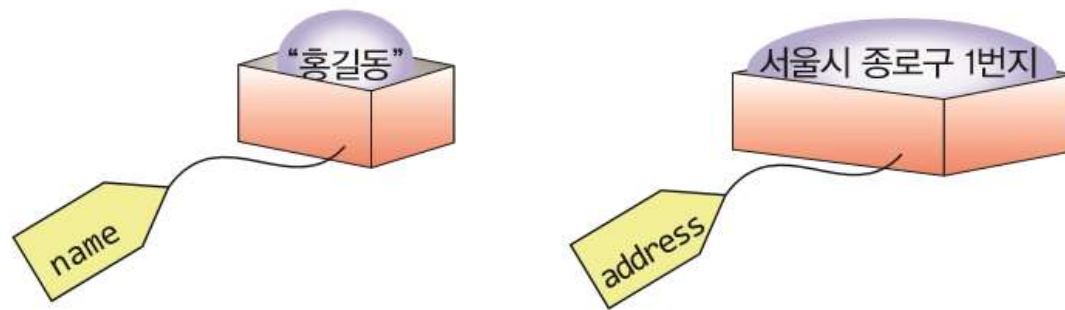
- 입문자들이 가장 많이 틀리는 문제 중의 하나가 =을 '양변이 같다'고 해석하는 것이다. 파이썬에서 = 기호는 "변수에 값을 저장하라"라는 의미이다. 혼동하지 않도록 하자.
- 등호는 ==와 같이 표시한다.



# 문자열도 변수에 저장할 수 있다!

- 파이썬의 변수에는 정수뿐만 아니라 문자열도 저장할 수 있다.

```
>>> name = "홍길동"  
>>> address = "서울시 종로구 1번지"
```



```
>>> print(name)  
홍길동  
>>> print(address)  
서울시 종로구 1번지
```



## 도전문제

무엇이 출력될까?

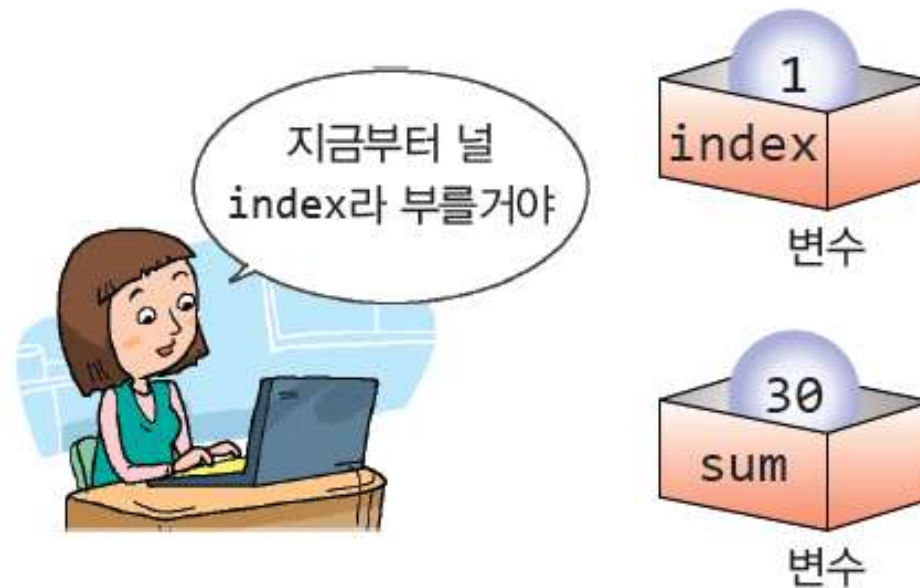
```
>>> x = 7  
>>> y = 6  
>>> print(x + y)
```

```
>>> x = '7'  
>>> y = '6'  
>>> print(x + y)
```



## ■ 변수의 이름

- 의미 있는 이름을 사용
- 소문자와 대문자는 서로 다르게 취급된다.
- 변수의 이름은 영문자와 숫자, 밑줄(\_)로 이루어진다.
- 변수의 이름 중간에 공백이 들어가면 안 된다. 단어를 구분하려면 밑줄(\_)을 사용 한다.





sum	# 영문 알파벳 문자로 시작
_count	# 밑줄 문자로 시작할 수 있다.
number_of_pictures	# 중간에 밑줄 문자를 넣을 수 있다.
King3	# 맨 처음이 아니라면 숫자도 넣을 수 있다.

2nd_base (X)	# 숫자로 시작할 수 없다.
money# (X)	# #과 같은 기호는 사용할 수 없다.



## ■ 낙타체

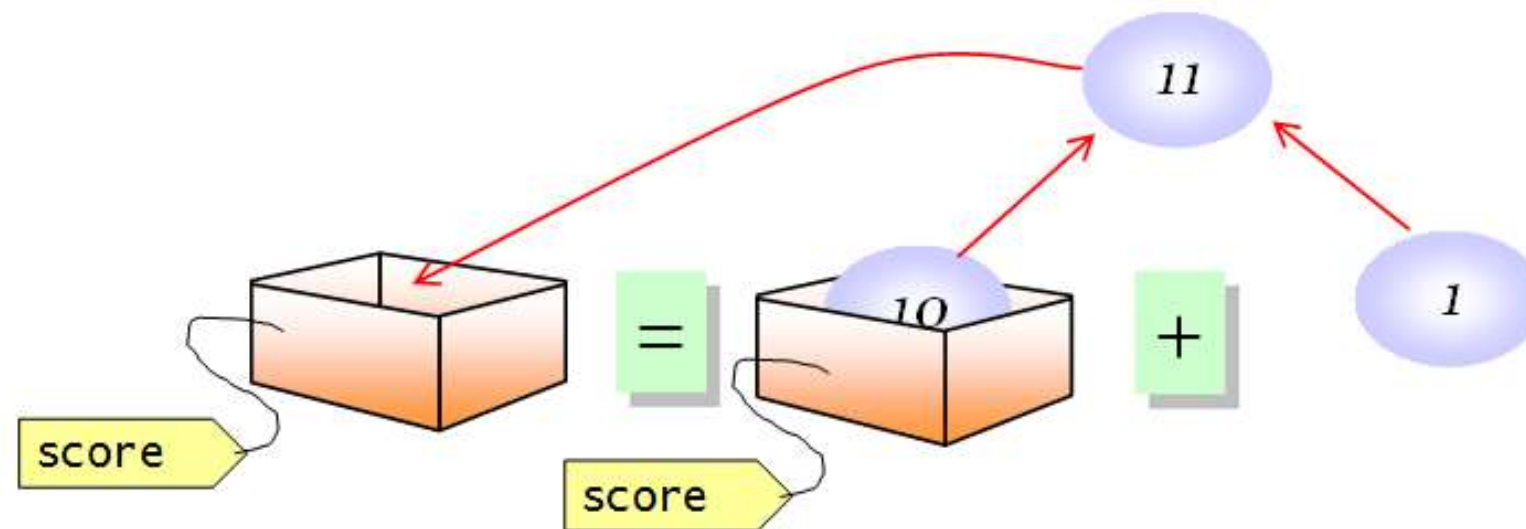
- 낙타체는 변수의 첫 글자는 소문자로, 나머지 단어의 첫 글자는 대문자로 적는 방법이다. 예를 들면, myNewCar처럼 첫 'm'은 소문자로, 나머지 단어들의 첫 글 자는 대문자로 표기한다





# 이런 것도 가능하다!

score = 10  
score = score + 1

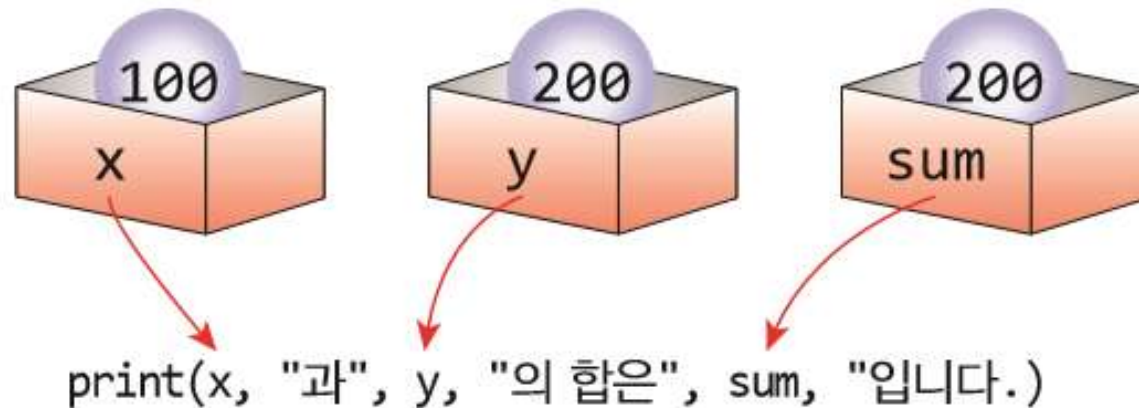




## 여러 값을 함께 출력하기

```
x = 100  
y = 200  
sum = x + y  
print(x, "과", y, "의 합은", sum, "입니다.")
```

100 과 200 의 합은 300 입니다.







## ■ 변수의 선언

- 변수는 어떤 값을 저장하기 위한 메모리 공간
- 가장 많이 사용하는 변수

```
boolVar, intVar, floatVar, strVar=True, 0, 0.0, ""
```

- type( ) 함수는 변수의 종류를 확인하는 함수

```
type(boolVar), type(intVar), type(floatVar), type(strVar)
```

출력 결과

```
(<class 'bool'>, <class 'int'>, <class 'float'>, <class 'str'>)
```

- 파이썬에서 변수의 데이터 형식은 값을 넣는 순간마다 변경될 수 있는 유연한 구조

```
myVar=100 → 정수형 변수를 생성함 (=국그릇 생성)
```

```
type(myVar) → <class 'int'>가 출력됨
```

```
myVar=100.0 → 이 순간에 실수형 변수로 변경됨 (=밥그릇으로 변경됨)
```

```
type(myVar) → <class 'float'>가 출력됨
```



# 사용자로부터 정수 입력받기

## input() 사용법

변수

사용자가 입력한 문자열을 숫자로 변환한다.

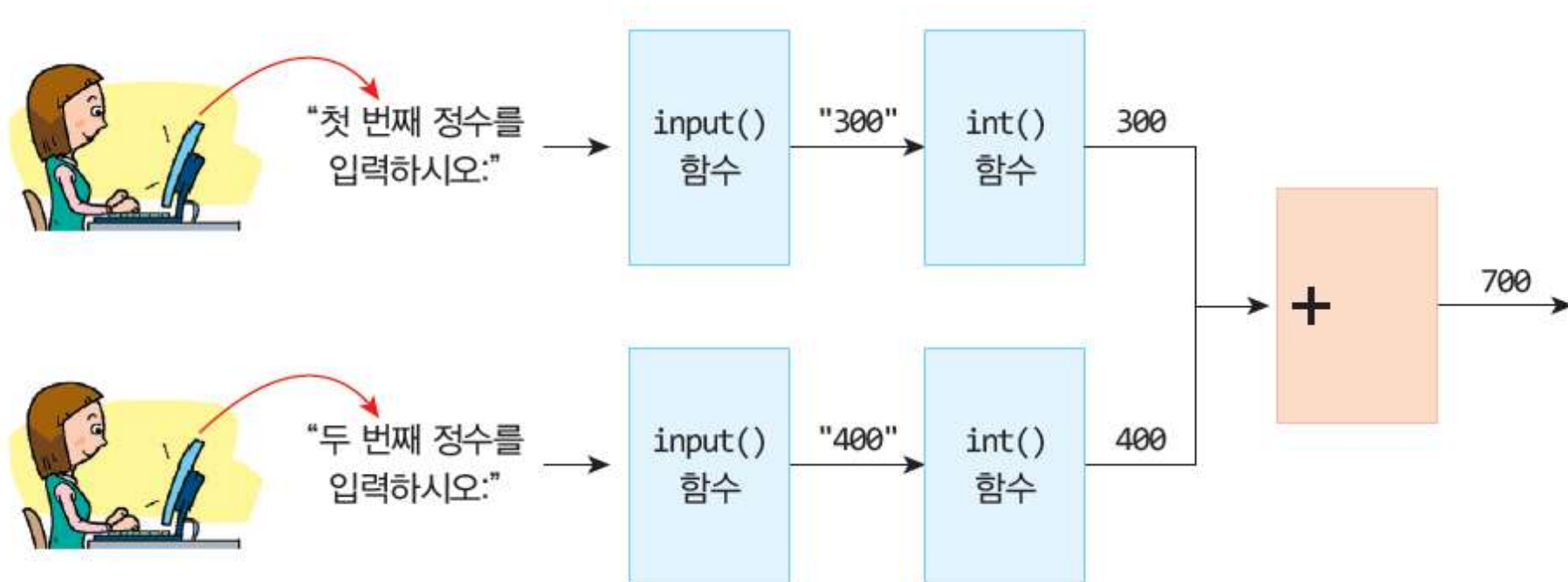
```
x = int(input("첫 번째 정수를 입력하시오: "))
```

안내 메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 받는다.



```
x = int(input("첫 번째 정수를 입력하시오: "))  
y = int(input("두 번째 정수를 입력하시오: "))  
sum = x + y  
print(x, "과", y, "의 합은", sum, "입니다.")
```

첫 번째 정수를 입력하시오: 300  
두 번째 정수를 입력하시오: 400  
300 과 400 의 합은 700 입니다.

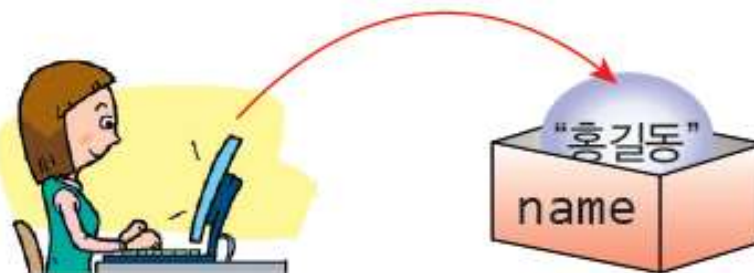




# 사용자로부터 문자열 입력받기

```
name = input("이름을 입력하시오: ")  
print(name, "씨, 안녕하세요?")  
print("파이썬에 오신 것을 환영합니다.")
```

이름을 입력하시오: 홍길동  
홍길동 씨, 안녕하세요?  
파이썬에 오신 것을 환영합니다.





## 도전문제

사용자의 이름을 물어보고 이어서 2개의 정수를 받아서 덧셈을 한 후에 결과를 출력하는 다음과 같은 프로그램을 작성해보자.

```
이름을 입력하시오: 홍길동
홍길동 씨, 안녕하세요?
파이썬에 오신 것을 환영합니다.
첫 번째 정수를 입력하시오: 300
두 번째 정수를 입력하시오: 400
300 과 400 의 합은 700 입니다.
```



- 서식을 지원하는 print( ) 함수 사용법
  - 서식은 앞에 %가 붙으며 %d는 정수(Decimal)를 의미

```
print("안녕하세요?")
```

➔ 안녕하세요

```
print("100")
```

```
print("%d" % 100)
```

➔ 글자 100(일영영)

➔ 숫자 100

```
print("100+100")
```

```
print("%d" % (100+100) )
```

➔ 글자 100+100

➔ 숫자 200



- 서식의 개수와 % 뒤에 나오는 숫자(또는 문자)의 개수가 같아야 함

```
print("%d" % (100, 200) )  
print("%d" %d" % (100) )
```

## → 오류발생

첫 번째 행에는 %d가 하나밖에 없는데 숫자는 두 개(100, 200)가 나왔고,  
두 번째 행에는 %d가 두 개인데, 숫자는 하나(100)밖에 나오지 않음

```
print ( "%d %d" % ( 100 , 200 ) )
```





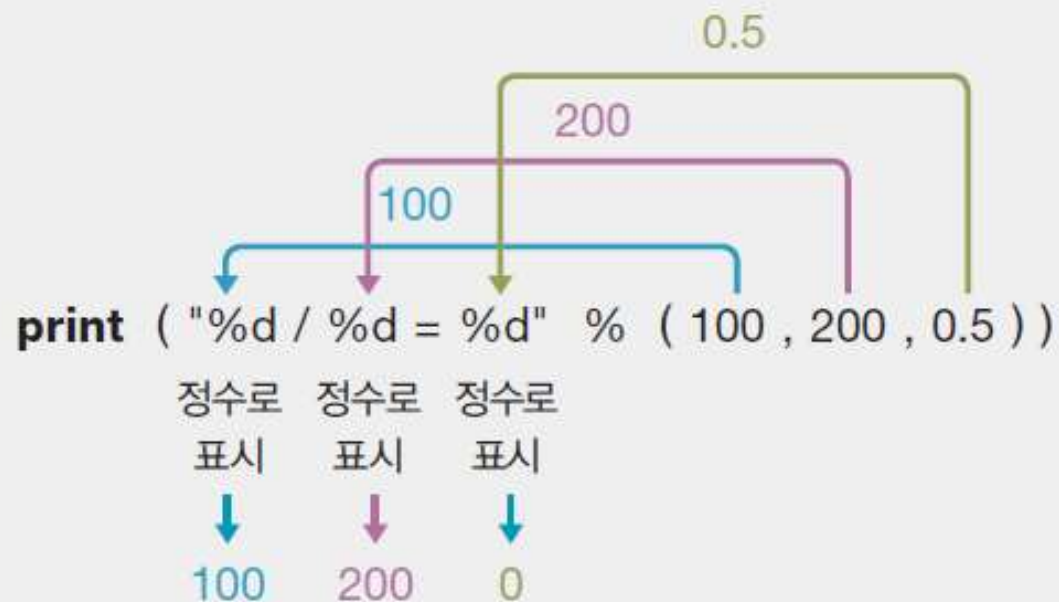
# print() 서식 지정

- 정수(%d) 외에 자주 사용되는 서식

```
print("%d / %d = %d" % (100, 200, 0.5))
```

➔  $100/200=0.5$  가 아닌  $100/200=0$  이 나옴.

세 번째 숫자 0.5는 실수(소수점이 있는 수)이지만 보여주는 방식이 정수임





# print() 서식 지정

서식	값의 예	설명
%d, %x, %o	10, 100, 1234	정수(10진수, 16진수, 8진수)
%f	0.5 , 1.0 , 3.14	실수(소수점이 붙은 수)
%c	"b", "한"	문자 한 글자
%s	"안녕", "abcdefg", "a"	한 글자 이상의 문자열

- 세 번째 %d 대신에 %f로 수정

```
print("%d / %d = %5.1f" % (100, 200, 0.5))
```



# print() 서식 지정

- 정수형 데이터 서식 지정

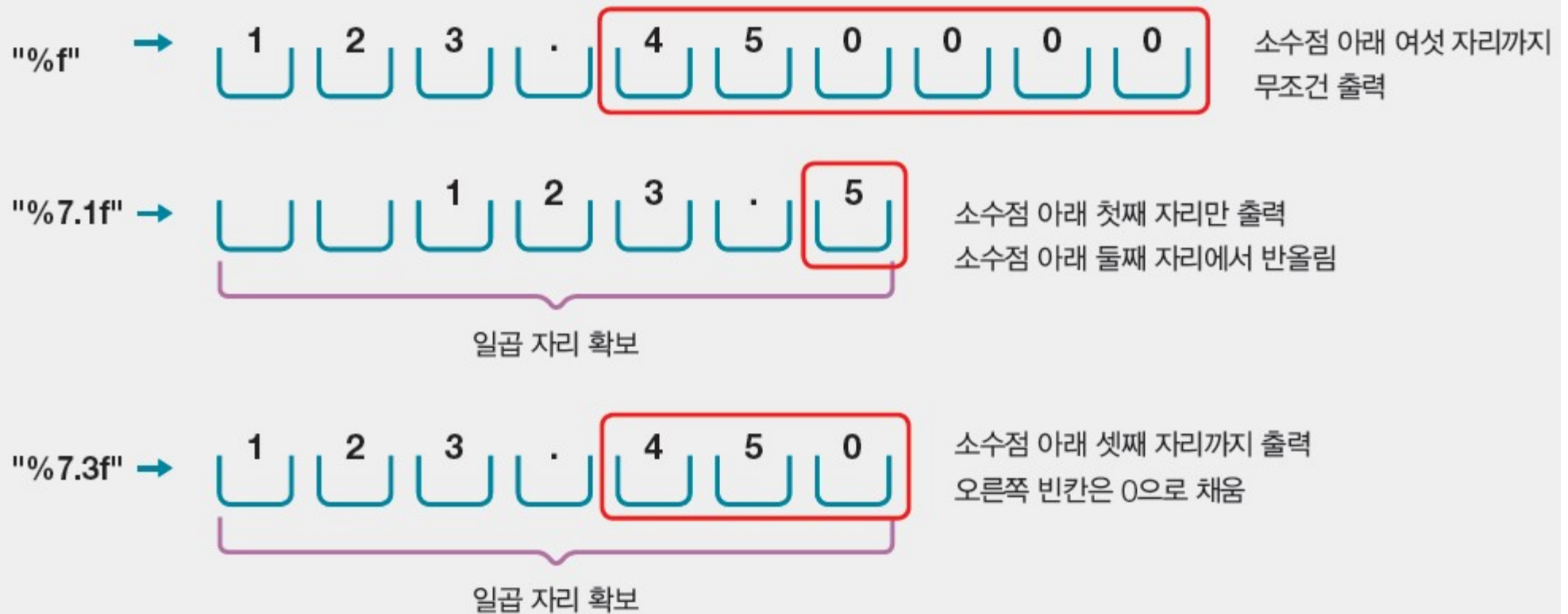
"%d" →  숫자의 자릿수만큼 정렬

"%5d" →  오른쪽에 붙여서 정렬  
다섯 자리 확보

"%05d" →  오른쪽에 붙여서 정렬  
빈칸을 0으로 채움  
다섯 자리 확보

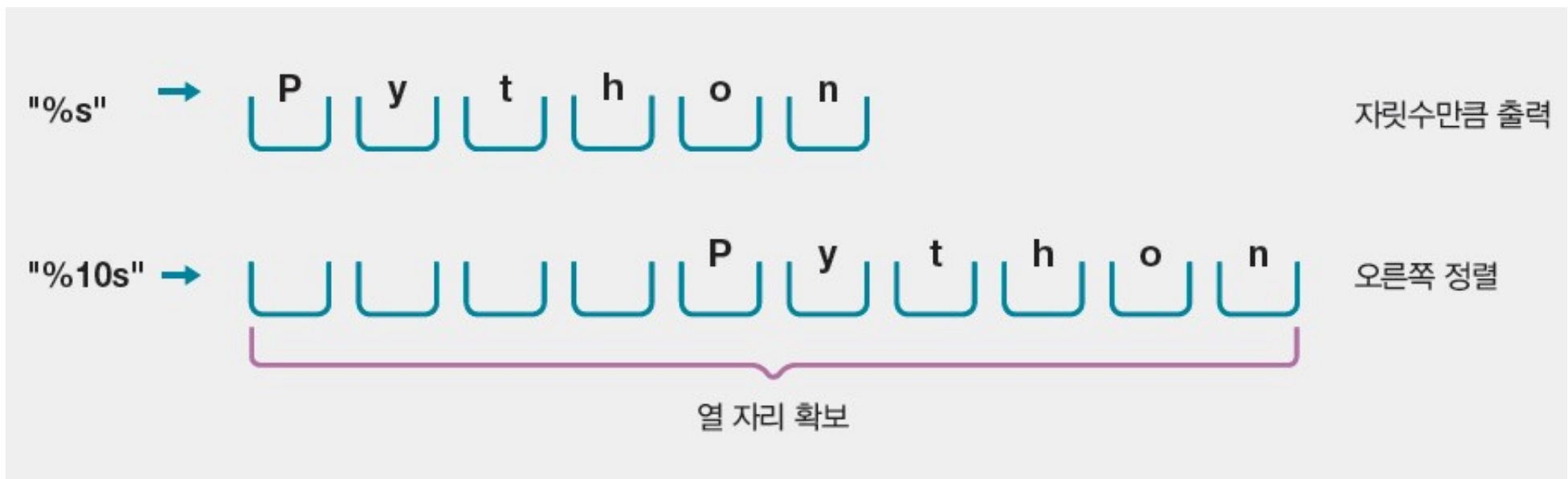


- 실수 형 데이터의 서식 지정
  - 두 번째 %7.1f는 소수점 을 포함한 전체 자리인 일곱 자리를 확보하고 소수점 아래는 한 자리만 차지한다는 의미





- 문자열 형 데이터 서식 지정



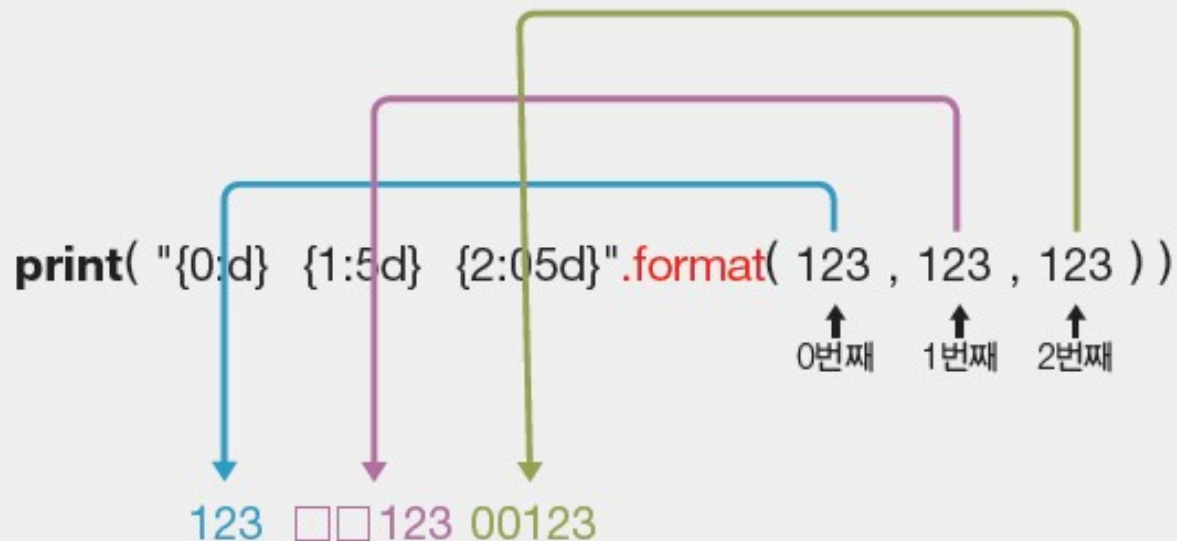


- format( ) 함수의 사용

```
print("%d %5d %05d" % (123, 123, 123))  
print("{0:d} {1:5d} {2:05d}".format(123, 123, 123))
```

➔ 두 행은 동일 결과를 출력.

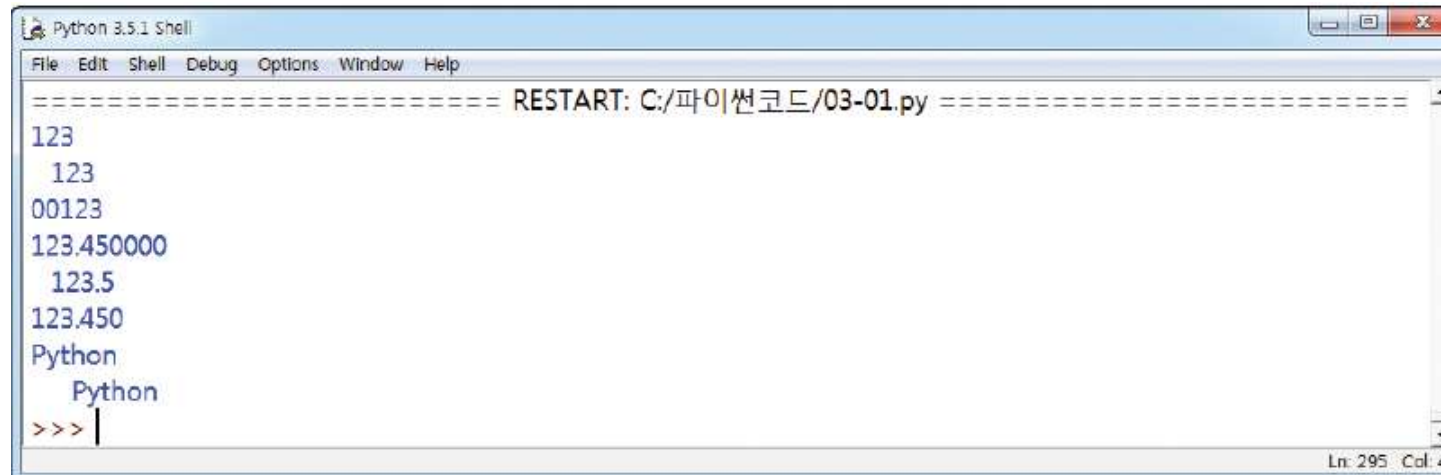
두 번째 행에서 { } 안의 0, 1, 2는 format( ) 안의 0번째, 1번째, 2번째 값에 대응한다는 의미. %d 에서 %를 떼고 d로 표시.





# print() 서식 지정

## ■ 서식 출력 연습



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/파이썬코드/03-01.py =====
123
  123
00123
123.450000
  123.5
123.450
Python
  Python
>>> |
```

```
print("%d" % 123)
print("%5d" % 123)
print("%05d" % 123)
```

```
print("%f" % 123.45)
print("%7.1f" % 123.45)
print("%7.3f" % 123.45)
```

```
print("%s" % "Python")
print("%10s" % "Python")
```



# print() 서식 지정

## ■ 다양한 이스케이프 문자

- print( )문은 내용을 출력한 후에 한 행을 넘겨줌

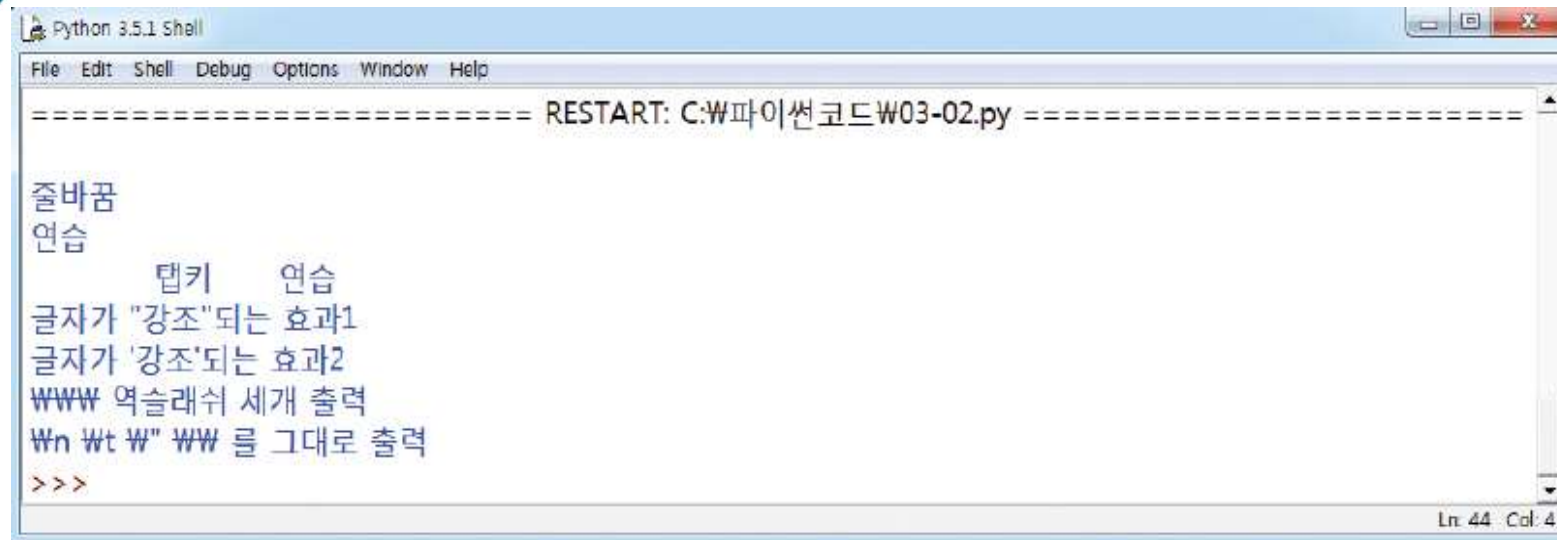
```
print("한행입니다. 또 한행입니다")  
print("한행입니다. \n또 한행입니다")
```

이스케이프 문자	역할	비고
\n	새로운 줄로 이동	<input type="button" value="Enter"/> 키를 누른 효과
\t	다음 탭으로 이동	<input type="button" value="Tab"/> 키를 누른 효과
\b	뒤로 한 칸 이동	<input type="button" value="Backspace"/> 키를 누른 효과
\\	\ 출력	
\'	' 출력	
\"	" 출력	





# print() 서식 지정



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\파이썬코드\W03-02.py =====

줄바꿈
연습
    탭키    연습
글자가 "강조"되는 효과1
글자가 '강조'되는 효과2
\\\\\\\\ 역슬래시 세개 출력
\\n \\t \\\" \\\\ 를 그대로 출력
>>>
```

```
print("\n줄바꿈\n연습 ")
print("\t탭키\t연습")
print("글자가 \"강조\"되는 효과1")
print("글자가 \'강조\'되는 효과2")
print("\\\\\\\\ 역슬래시 세개 출력")
print(r"\n \t \" \\\\ 를 그대로 출력")
```



## 2, 10, 16 진수

### ■ 진수

10진수(0~9)	2진수(0~1)	16진수(0~F)	10진수(0~9)	2진수(0~1)	16진수(0~F)
00	0000	0	08	1000	8
01	0001	1	09	1001	9
02	0010	2	10	1010	A
03	0011	3	11	1011	B
04	0100	4	12	1100	C
05	0101	5	13	1101	D
06	0110	6	14	1110	E
07	0111	7	15	1111	F

- 16진수는 0~F까지 총 16가지의 숫자로 표현. 16진수가 필요한 이유는 2진수의 네 자리와 16진수 한 자리가 맞아떨어지기 때문임



## 2, 10, 16 진수

- 파이썬에서는 2진수를 입력하려면 앞에 0b 또는 0B를 붙여줌. 출력 시 10진수로 나옴.

```
0b10010011
```

출력 결과

```
147
```

- 또는 int('숫자', 진수)를 사용하면 10진수로 변환

```
int('10010011', 2)
```

출력 결과

```
147
```

- 16진수는 앞에 0x 또는 0X를 붙여주거나 int() 함수를 사용해 10진수로 변환.

```
0x93 ; int('93', 16)
```

출력 결과

```
147
```

```
147
```



## 2, 10, 16 진수

- hex(숫자), oct(숫자), bin(숫자) 함수
  - hex()는 16진수로, oct()는 8진수로, bin()은 2진수로 결과 출력

```
bin(13) ; bin(0x13) ; bin(0xC5F7)
```

출력 결과

```
'0b1101'
```

```
'0b10011'
```

```
'0b1100010111110111'
```



- 사용자에게 경기장, 점수, 이긴 팀, 진 팀, 우수 선수를 질문하고 변수에 저장한다. 이들 문자열에 문장을 붙여서 기사를 작성한다.

경기장은 어디입니까?서울  
이긴팀은 어디입니까?삼성  
진팀은 어디입니까?**LG**  
우수선수는 누구입니까?홍길동  
스코어는 몇대몇입니까?8:7

=====

오늘 서울 에서 야구 경기가 열렸습니다.  
삼성과 **LG** 은 치열한 공방전을 펼쳤습니다.  
홍길동이 맹활약을 하였습니다.  
결국 삼성이 **LG** 를 8:7 로 이겼습니다.

=====



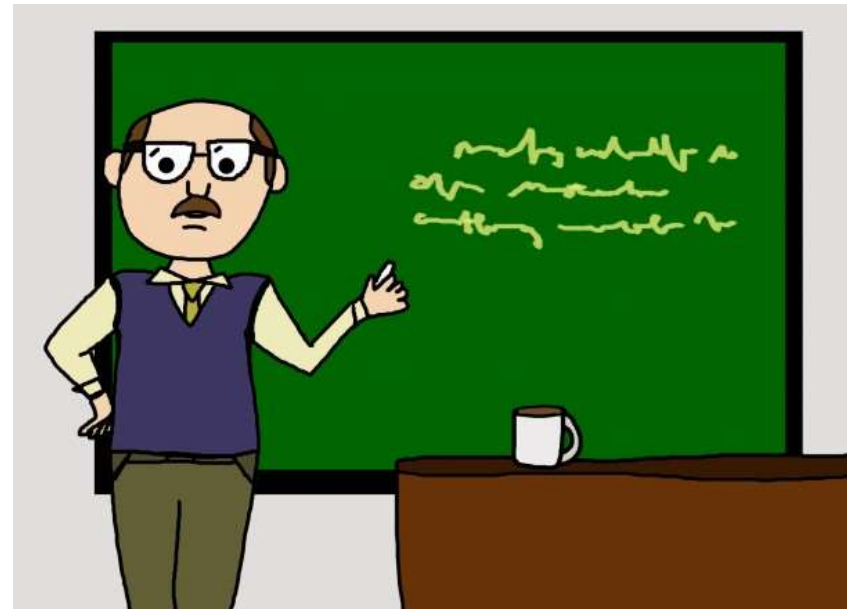
```
# 사용자의 대답을 변수에 저장한다.
stadium = input("경기장은 어디입니까?")
winner = input("이긴팀은 어디입니까")
loser = input("진팀은 어디입니까?")
vip = input("우수선수는 누구입니까?")
score = input("스코어는 몇대몇입니까?")

# 변수와 문자열을 연결하여 기사를 작성한다.
print("")
print("=====")
print("오늘", stadium, "에서 야구 경기가 열렸습니다.")
print(winner, "과", loser, "은 치열한 공방전을 펼쳤습니다.")
print(vip, "이 맹활약을 하였습니다.")
print("결국", winner,"가", loser,"를 ", score,"로 이겼습니다.")
print("=====
```



## 이번 장에서 배운 것

- 컴퓨터에서는 변수를 사용하여 어떤 것을 컴퓨터 메모리 안에 저장할 수 있다.
- 변수들은 이름을 가지고 있다.
- 변수들은 숫자뿐만 아니라 문자열도 저장할 수 있다. 사실은 어떤 것이든지 저장이 가능하다.





**THANK**  

---

**YOU**