

# 새내기 파이썬



## 2장 변수와 자료형



# 하스 기 모 표 기

- 변수를 선언하여서 숫자나 문자열을 저장할 수 있나요?
- 사용자로부터 입력을 받아서 변수에 저장할 수 있나요?
- 문자열을 합할 수 있나요?
- 자료형을 설명할 수 있나요?



## 2장에서 만든 프로그램

첫 번째 정수를 입력하시오: 10  
두 번째 정수를 입력하시오: 2  
 $10 + 2 = 12$   
 $10 - 2 = 8$   
 $10 * 2 = 20$   
 $10 / 2 = 5$

경기장은 어디입니까?서울  
이긴팀은 어디입니까?타이거  
진팀은 어디입니까?라이온스  
우수선수는 누구입니까?홍길동  
스코어는 몇대몇입니까?8:7

=====

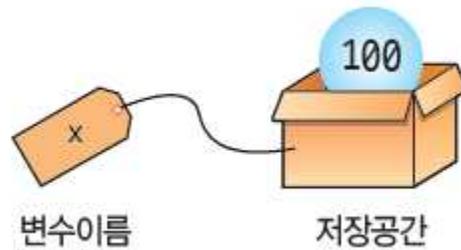
오늘 서울에서 야구 경기가 열렸습니다.  
타이거와 라이온스는 치열한 공방전을 펼쳤습니다.  
홍길동이 맹활약을 하였습니다.  
결국 타이거가 라이온스를 8:7로 이겼습니다.

=====



# 변수

- 변수(variable)는 컴퓨터의 메모리 공간에 이름을 붙이는 것으로 우리는 여기에 값을 저장할 수 있다.



변수는 이름 붙인 메모리 공간으로  
우리는 여기에 값을 저장할 수 있습니다.



## 변수 정의하기

- 파이썬에서는 변수에 값을 저장하면 변수가 자동으로 생성된다.

변수 = 값

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



# 변수에 저장된 값을 변경할 수 있다!

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

```
>>> x
```

```
>>> x = 200
```

```
x
```

```
200
```



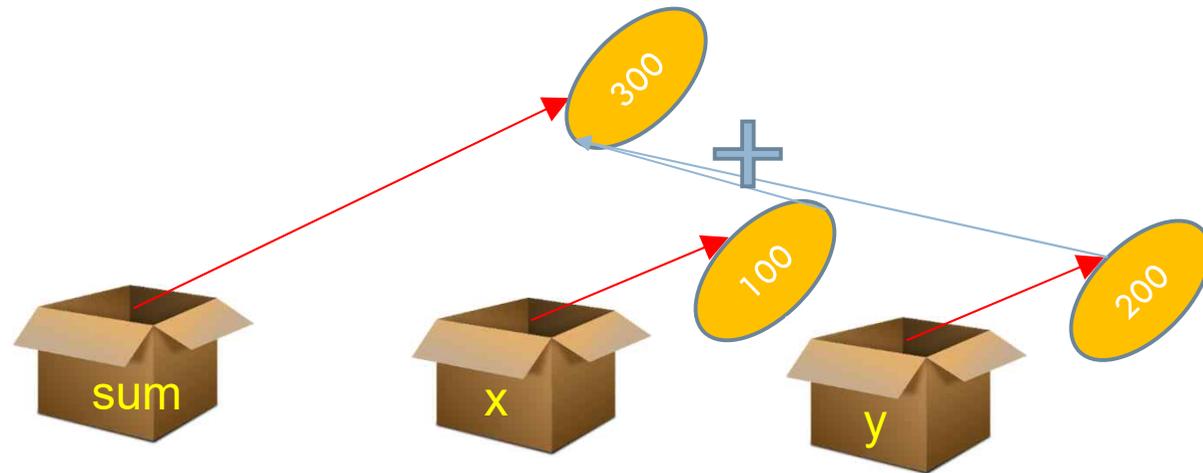


## 예제: 두 수의 합 계산하기

```
x = 100  
y = 200  
sum = x + y  
print("합은", sum)
```

```
# 변수 x를 생성하고 100을 저장한다.  
# 변수 y를 생성하고 200을 저장한다.  
# 변수 sum을 생성하고 x+y를 저장한다.
```

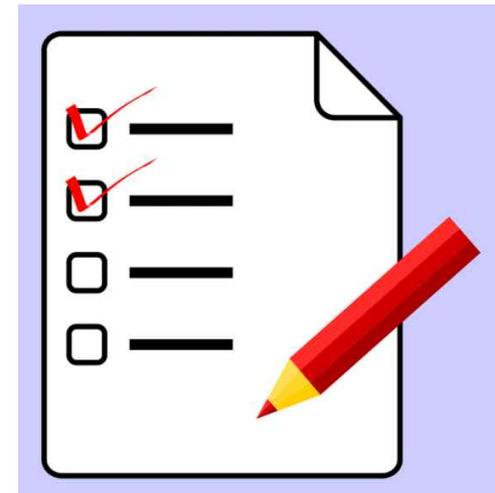
합은 300





## 중간점검

1. 변수를 다시 설명해보자.
2. 변수  $a$ 에 200을 변수  $b$ 에 100을 저장한 후에  $(a-b)$ 하여 결과를 출력하는 프로그램을 작성해보자.
3. 밑변이 10이고 높이가 10인 삼각형의 면적을 계산하는 프로그램을 작성해보자.





## 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.





# 파이썬의 예약어

- 예약어란 파이썬이 사용하는 특수한 단어이다. 파이썬의 예약어는 다음과 같다.

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	



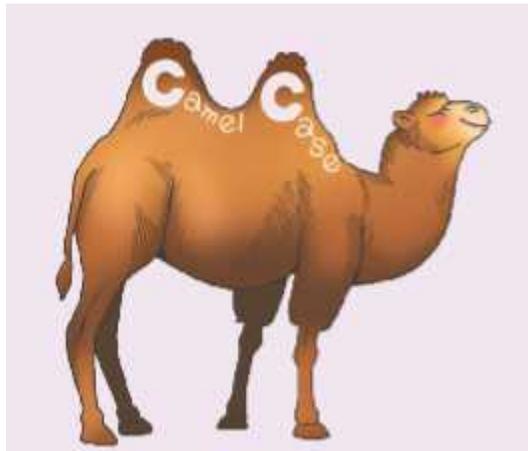
# 변수의 이름

변수 이름	설명
size	가능하다.
cloud9	가능하다. 변수는 영문자, 숫자, _로 이루어진다.
max_size	가능하다. 변수의 중간에 _가 있어도 된다.
_count	가능하다. _가 앞에 붙으면 클래스 내부에서만 사용하는 변수라는 의미도 있다.
6pack	올바르지 않다! 숫자가 앞에 오면 안된다.
mid score	올바르지 않다! 중간에 공백이 있으면 안된다.
class	올바르지 않다! 예약어를 변수의 이름으로 사용할 수 없다.
money#	올바르지 않다! 기호를 변수의 이름으로 사용하면 안 된다.



## 낙타체

- 낙타체는 변수의 첫 글자는 소문자로, 나머지 단어의 첫 글자는 대문자로 적는 방법이다. 예를 들면, `myNewCar`처럼 첫 'm'은 소문자로, 나머지 단어들의 첫 글자는 대문자로 표기한다





# 예제: 원의 면적 계산하기

반지름 20인 원의 면적 = 1256.0

## 알고리즘

STEP #1. 사용자로부터 원의 반지름을 입력받는다.

STEP #2. 공식을 적용하여 면적을 계산한다.

$$\text{area} = \text{radius} * \text{radius} * \pi$$

STEP #3. 면적을 화면에 출력한다.

## 코드

```
# 변수 radius에 값을 저장한다.  
radius = 10  
  
# 공식을 적용하여 면적을 계산한다  
area = 3.14 * radius * radius  
  
# 면적을 화면에 출력한다.  
print("반지름", radius, "인 원의 면적=", area)
```



# 장가저거 중간점검

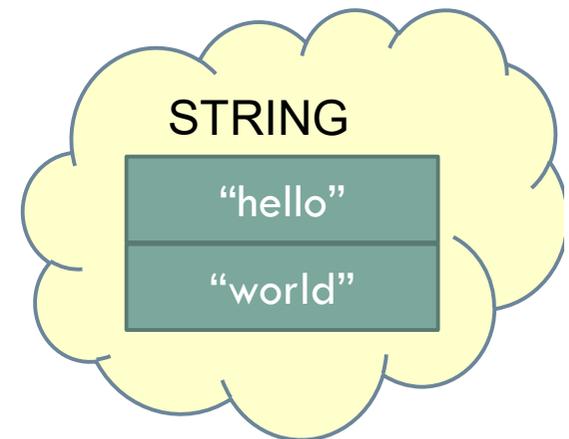
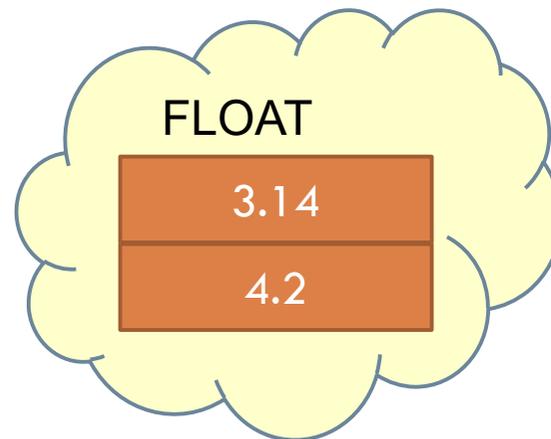
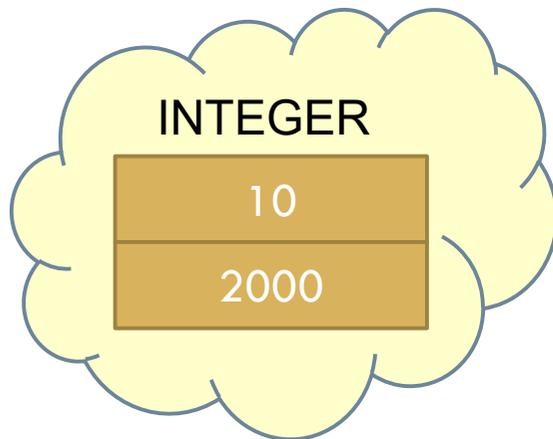
1. 변수 이름의 첫 번째 글자로 허용되는 것은 무엇인가?
2. 파이썬에서 고유한 의미를 가지고 있는 단어들을 무엇이라고 하는가?
3. 파이썬에서는 대소문자를 구분하는가?





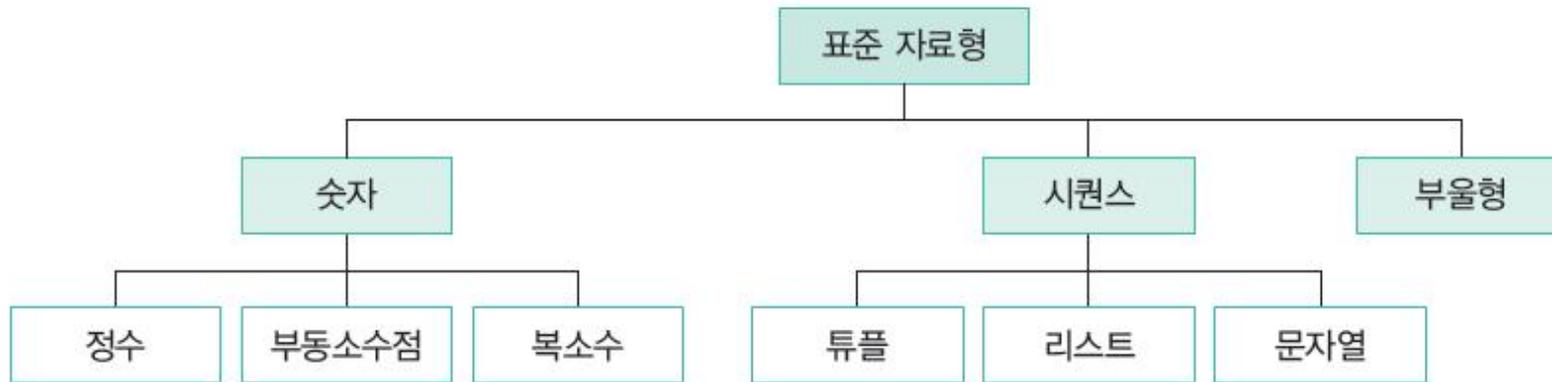
# 자료형(데이터 타입)

- 자료형은 **데이터의 타입**(type, 유형)이다. 프로그래밍에서 자료형은 중요한 개념이다. 반드시 정해진 타입의 데이터만 저장되고 처리된다.
- 변수는 특정한 자료형의 데이터를 저장할 수 있으며, 자료형에 따라서 허용되는 연산이 달라진다.





# 파이썬의 표준 자료형



자료형	예
정수(int)	..., -2, -1, 0, 1, 2, ...
부동소수점수(float)	3.2, 3.14, 0.12
문자열(str)	'Hello World!', "123"



# 자료형을 알려면?

```
>>> type(1234)          # int 형  
<class 'int'>
```

```
'blue'    → <class 'str'>  
23        → <class 'int'>  
3.14     → <class 'float'>  
2 < 3    → <class 'bool'>  
(2, 3, 4) → <class 'tuple'>
```





# 다양한 진법 가능

- 정수형은 소수점이 없는 수로 1, -23, 0 등이 여기에 속한다.

```
a = 0xFF          # 16진수
B = 0o77          # 8진수
C = 0b1111       # 2진수

print(a, b, c)    # 모든 변수를 10진수로 출력한다.
```

```
255 63 15
```

Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111



# 10진법, 8진법, 16진법

- 8진법
  - $0o12_8 = 1 \times 8^1 + 2 \times 8^0 = 10$
- 16진법
  - $0xA_{16} = 10 \times 16^0 = 10$

10진수	8진수	16진수
0	00	0x0
1	01	0x1
2	02	0x2
3	03	0x3
4	04	0x4
5	05	0x5
6	06	0x6
7	07	0x7
8	010	0x8
9	011	0x9
10	012	0xa
11	013	0xb
12	014	0xc
13	015	0xd
14	016	0xe
15	017	0xf
16	020	0x10
17	021	0x11
18	022	0x12



# 부동소수점형

- 부동소수점형은 실수를 나타내는 자료형이다. 부동소수점이란 컴퓨터 안에서 실수를 표현하기 위한 하나의 방법이다.

```
a = 3.14  
b = 1.23e2  
print(a, b)
```

↑는 지수부를 의미한다.  $1.23e2$ 는  $1.23 \times 10^2$

```
3.14 123.0
```

```
a = 3.14  
b = 7.12  
print(a+b, a-b, a*b, a/b)
```

```
10.26 -3.98 22.3568 0.4410112359550562
```



# 문자열

- 문자열(string)은 문자들의 시퀀스(sequence of characters)이다. 문자들이 실로 연결된 형태를 상상하면 된다.



문자열은 문자들의 나열입니다.



# 문자열

출력 시에는 항상 작은 따옴표를 이용한다.

```
>>> greeting="Merry Christmas!"  
>>> greeting  
'Merry Christmas!'
```

```
>>> greeting='Happy New Year!'  
>>> greeting  
'Happy New Year!'
```

문자열 안에 따옴표를 넣을 때는 다른 형태의 따옴표를 사용한다.

```
>>> message="철수가 '안녕'이라고 말했습니다."  
>>> message  
철수가 '안녕'이라고 말했습니다.
```

```
>>> greeting=""지난 한해 저에게 보여주신 보살핌과 사랑에  
깊은 감사를 드립니다.  
새해에도 하시고자 하는 일  
모두 성취하시기를 바랍니다.""  
>>> greeting  
지난 한해 저에게 보여주신 보살핌과 사랑에  
깊은 감사를 드립니다.  
새해에도 하시고자 하는 일  
모두 성취하시기를 바랍니다.
```

여러 줄로 된 문자열은 ''' ... '''을 사용한다



문자열은 +로 합칠 수 있다.

```
>>> 'Harry ' + 'Porter'  
'Harry Porter'
```

```
>>> first_name="길동"  
>>> last_name="홍"  
>>> name = last_name+first_name  
>>> name  
홍길동
```

문자열은 + 연산자로 합칠 수 있다.



# 문자열 <-> 숫자

```
>>> "Student"+26
Traceback (most recent call last):
  File "<pyshell#16>", line 1, in <module>
    "Student"+26
TypeError: can only concatenate str (not "int") to str
```

```
>>> "Student"+str(26)
'Student26'
```

str()은 숫자를 문자열로 바꾼다.

```
>>> price = int("123")
>>> height = float("3.14");
```

int()은 문자열을 정수로, float()은 문자열을 실수로 바꾼다.





## 문자열 메소드

- `len()`: 문자열의 길이를 계산한다.
- `upper()`와 `lower()` 함수는 문자열을 대문자나 소문자로 바꾼다.
- `find()`: 문자열에서 어떤 단어를 찾는다. 위치는 0부터 시작하는 인덱스로 반환된다.

```
>>> message = "Merry Christmas!"
>>> len(message)           # 문자열의 길이를 반환한다.
16
>>> message.upper()       # 문자열을 대문자로 만들어서 반환한다.
'MERRY CHRISTMAS!'
>>> message.find("Ch")    # 문자열 안에서 "Ch"를 찾는다.
6
```



# 문자열 안의 문자에 접근하기

- 문자열은 여러 개의 문자로 이루어진다. 문자열 안의 문자들은 다음과 같은 위치를 가지게 된다.

<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
<i>P</i>	<i>y</i>	<i>t</i>	<i>h</i>	<i>o</i>	<i>n</i>
<i>-6</i>	<i>-5</i>	<i>-4</i>	<i>-3</i>	<i>-2</i>	<i>-1</i>

```
>>> s = "Python"
>>> s[0]          # 첫 번째 문자
'P'
>>> s[1]          # 두 번째 문자
'y'
>>> s[-1]         # 마지막 문자
'n'
```



## 문자열 안에 변수 출력

- 문자열에 변수의 값을 삽입하여 출력하고 싶으면 많은 방법이 있다. 저자가 추천하는 방법은 f-문자열(f-string)이다. 이 방법은 문자열 안에 출력하고 싶은 변수를 중괄호로 감싸서 넣는 방법이다.

```
>>> price = 10000
>>> print(f"상품의 가격은 {price}원입니다.")
상품의 가격은 10000원입니다.
```



# 문자열 안에 변수 출력

수식도 출력할 수 있다.

```
product = "coffee"  
count = 3  
price = 10000  
print(f"상품 {product} {count}개의 가격은 {count*price}원입니다.")
```

상품 coffee 3개의 가격은 30000원입니다.

```
pi = 3.141592  
print(f"원주율={pi:.2f}") # 소수점 두번째 자리까지 출력
```

원주율=3.14

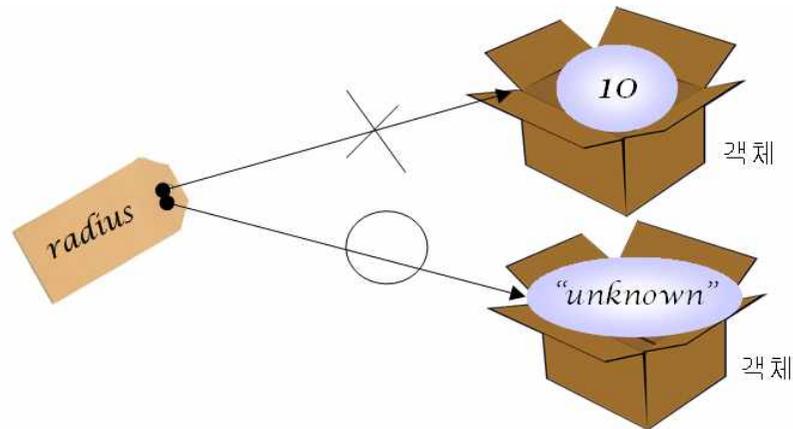


# 변수의 세부 구현 사항

- 파이썬에서는 동일한 변수에 여러 가지 자료형의 데이터를 저장할 수 있다.

```
>>> radius = 10
>>> radius = 3.14
>>> radius = "unknown"
```

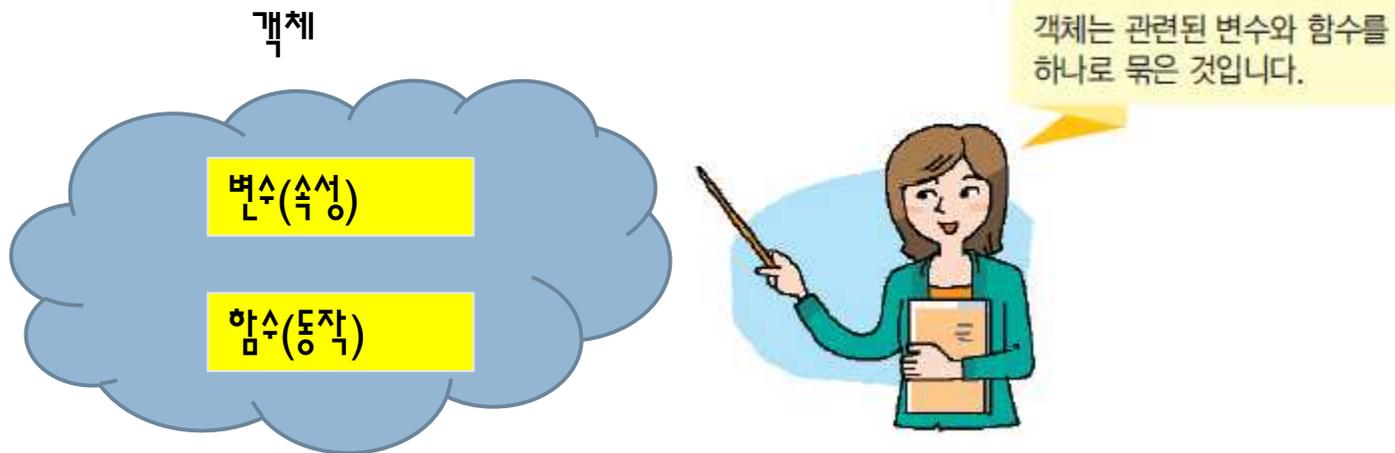
- 파이썬이 변수에 어떤 자료형의 데이터든지 저장할 수 있는 것은, 파이썬에서는 모든 것이 객체(object)로 되어 있기 때문이다.





# 객체

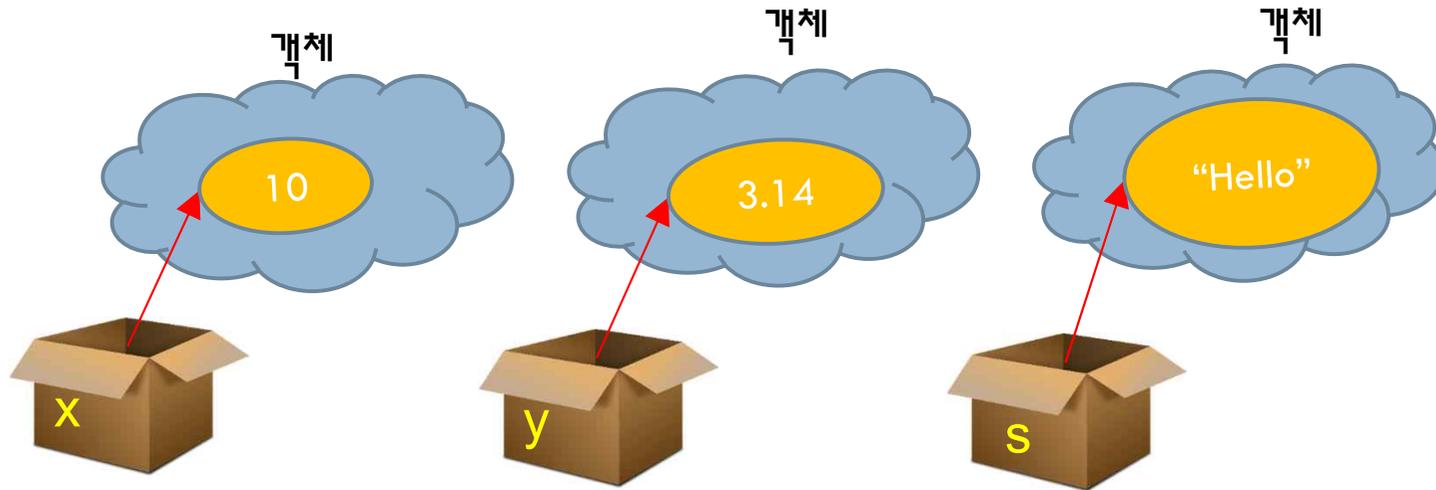
- 객체(object)란 프로그래밍에서 관련 있는 변수와 함수(아직 학습하지 않았지만 어떤 동작이라고 생각하자)를 하나로 묶은 것이다.





사실 파이썬에서는 모든 것이 객체이다.

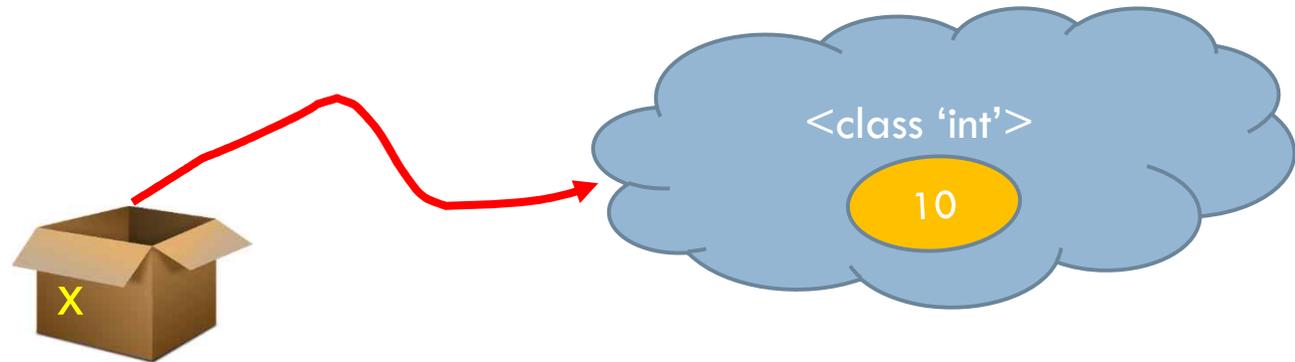
- 파이썬에서는 정수나 실수도 객체로 저장된다.





# 파이썬에서는 모든 것이 객체

- 객체(object)는 속성과 동작을 하나의 덩어리로 생각하여 프로그래밍 하는 기술이다. 파이썬에서는 정수부터 시작하여 모든 것이 객체로 구현되어 있다.
- 정수 10의 자료형 <class 'int'>는 10이 정수 클래스라는 것을 나타낸다. 클래스는 객체를 찍어내는 틀이다.

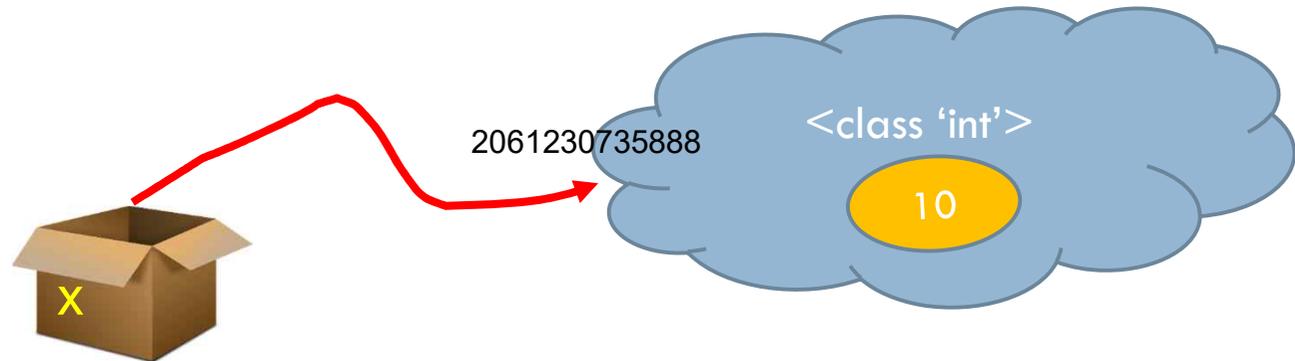




# 객체의 참조값을 알려면?

```
>>> x = 10  
>>> id(x)  
2061230735888
```

참조값이란 객체를 구별할 수 있는 값이다. 일반적으로 객체가 저장된 주소가 사용된다.





# 변수의 세부 구현 사항

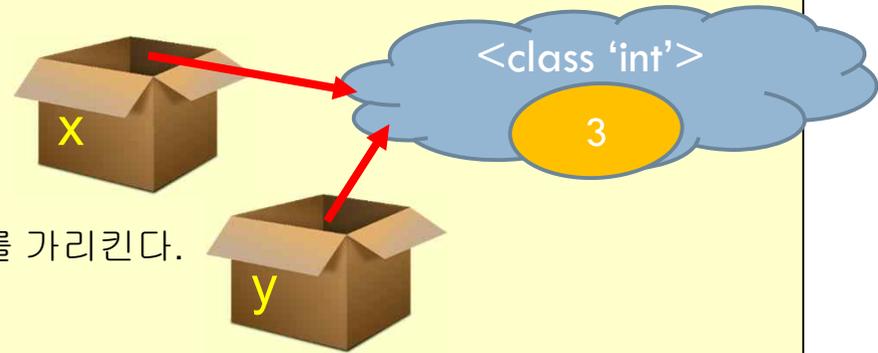
```
>>> x = 3  
>>> y = x
```

# 변수 y에 변수 x의 참조값이 복사된다.

```
>>> id(x)  
140721955779472
```

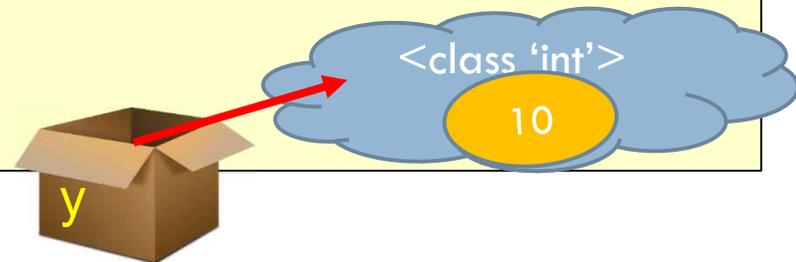
```
>>> id(y)  
140721955779472
```

# 같은 주소를 가리킨다.



```
>>> y = 10  
>>> id(y)  
140721955779696
```

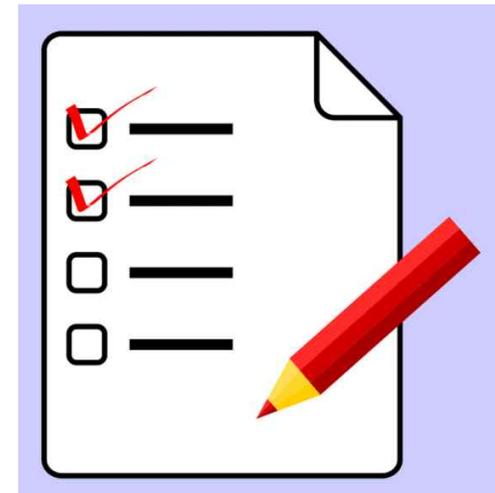
# 변수 y에 새로운 값이 할당되면 주소가 달라진다.





## 중간 점검

1. 파이썬에는 어떤 기본 자료형이 있는가? 3가지만 말해보자.
2. 변수의 자료형을 알려면 어떤 함수를 사용하는가?
3. `id()` 함수가 반환하는 것은 무엇인가?
4. 파이썬 변수에 정수를 저장하였다가 실수를 저장하는 것도 가능한가? 그 이유는 무엇인가?



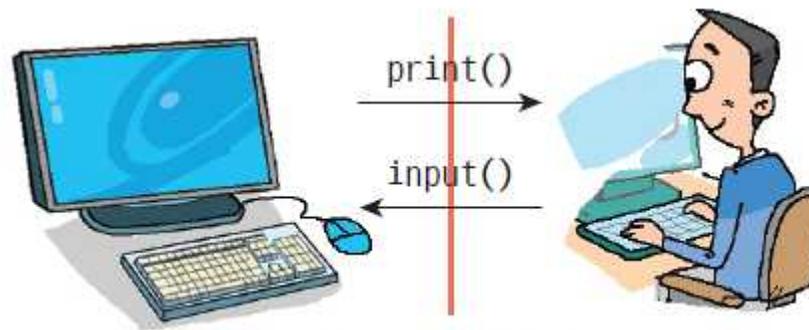


# input() 함수

- input()은 전달된 메시지를 화면에 출력하고 사용자의 입력을 기다린다. 사용자가 입력하면 input()은 사용자의 입력을 문자열 형태로 반환하고 종료한다.

변수 = input("값을 입력하시오")

안내 메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 반환한다.



사용자 인터페이스



# input() 함수

```
>>> name = input("이름을 입력하시오: ")
```

```
이름을 입력하시오: 홍길동
```

```
>>> name
```

```
홍길동
```

```
>>> print(name, "씨, 안녕하세요?")
```

```
홍길동 씨, 안녕하세요?
```



## 예제: 정수의 합 계산

```
x = input("첫 번째 정수를 입력하시오:")  
y = input("두 번째 정수를 입력하시오:")  
sum = x + y  
print("합은 ", sum)
```

```
첫 번째 정수를 입력하시오: 300  
첫 번째 정수를 입력하시오: 400  
합은 300400
```

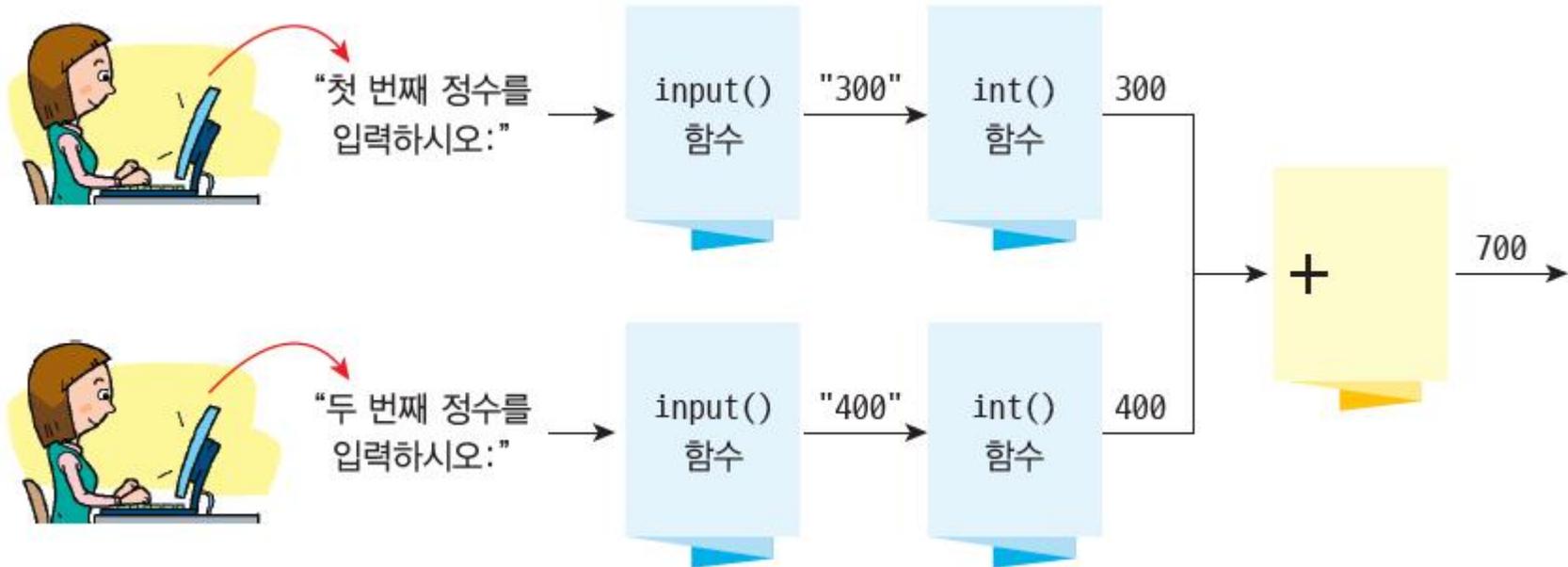
문자열을 정수로 바꾸지 않았다!

```
x = int(input("첫 번째 정수를 입력하시오:")) # 문자열을 정수로 변환한다.  
y = int(input("두 번째 정수를 입력하시오:")) # 문자열을 정수로 변환한다.  
sum = x + y  
print("합은 ", sum)
```

```
첫 번째 정수를 입력하시오: 300  
첫 번째 정수를 입력하시오: 400  
합은 700
```



# 예제: 정수의 합 계산

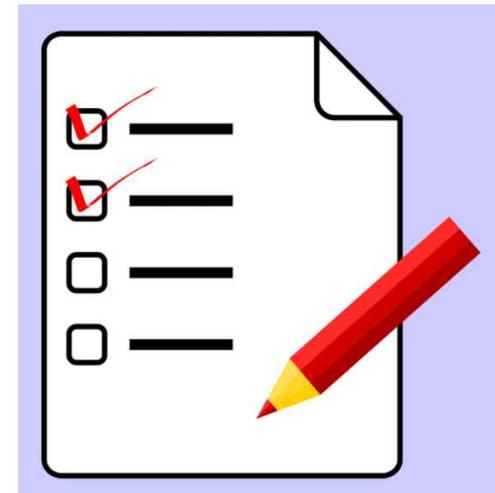




# 직가저거 중간점검

1. 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 700입니다.





# Lab: 간단한 계산기

첫 번째 정수를 입력하시오: 10

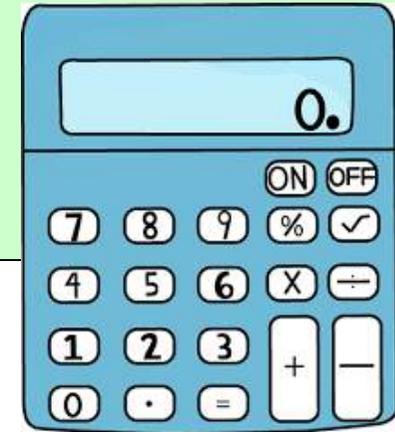
두 번째 정수를 입력하시오: 2

$$10 + 2 = 12$$

$$10 - 2 = 8$$

$$10 * 2 = 20$$

$$10 / 2 = 5.0$$





# Sol: 간단한 계산기

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

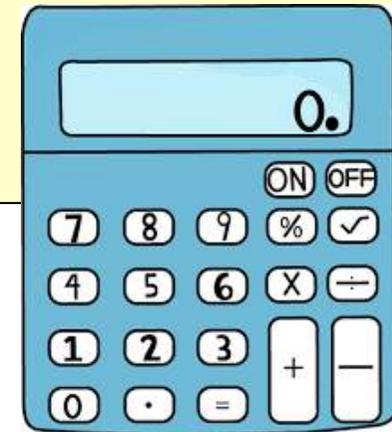
```
# 문자열을 정수로 변환한다.  
# 문자열을 정수로 변환한다.
```

```
result = x + y  
print(x, "+", y, "=", result)
```

```
result = x - y  
print(x, "-", y, "=", result)
```

```
result = x * y  
print(x, "*", y, "=", result)
```

```
result = x / y  
print(x, "/", y, "=", result)
```





## 주석

- 주석(comment)은 소스 코드에 붙이는 설명글과 같은 것이다. 주석은 프로그램이 하는 일을 설명한다. 주석은 프로그램의 실행 결과에 영향을 끼치지 않는다.

```
##  
#           이 프로그램은 화씨 온도를 받아서 섭씨 온도로 변환한다.  
#  
ftemp = 100                                # 화씨 온도 100를 변수에 저장한다.  
  
ctemp = (ftemp-32.0)*5.0/9.0              # 화씨온도->섭씨온도  
print("섭씨온도:", ctemp)                 # 섭씨온도를 화면에 출력한다.
```



## 주석

- 만약 코드 중에서 실행하고 싶지 않은 문장이 있다면 앞에 # 기호를 놓으면 주석으로 처리된다.

```
##  
#         이 프로그램은 정수들의 합을 계산한다.  
#  
x = 100  
y = 200  
sum = x + y  
#diff = x - y  
print("합은 ", sum)
```

합은 300



# 상수

- 상수는 한번 값이 결정되면 절대로 변경되지 않는 변수를 의미한다



변수



상수

그림 3. 변수와 상수



# 상수

파이썬에서는 통상적으로 대문자로 상수를 표현한다.  
그러다 고 변경 안되는 것은 아니다.

```
TAX_RATE = 0.35  
PI = 3.141592  
MAX_SIZE = 100
```

```
INCOME = 1000  
TAX_RATE = 0.35  
  
tax = INCOME * TAX_RATE  
net_income = INCOME - tax  
...
```

소득과 소득세율이 변경되어 다면 상수의 정의만  
변경하며 된다.



# Lab: 로봇 기자 만들기

- 기사의 틀을 만들어두고, 핵심 요소를 변수로 만들면 자동으로 기사를 작성할 수 있다.

경기장은 어디입니까?서울  
이긴팀은 어디입니까?타이거  
진팀은 어디입니까?라이온스  
우수선수는 누구입니까?홍길동  
스코어는 몇대몇입니까?8:7

=====  
오늘 서울에서 야구 경기가 열렸습니다.  
타이거와 라이온스는 치열한 공방전을 펼쳤습니다.  
홍길동이 맹활약을 하였습니다.  
결국 타이거가 라이온스를 **8:7**로 이겼습니다.  
=====



# Sol: 로봇 기자 만들기

- 기사의를 틀을 만들어두고 핵심 요소는 변수로 만들면 자동으로 기사를 작성할 수 있다.

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

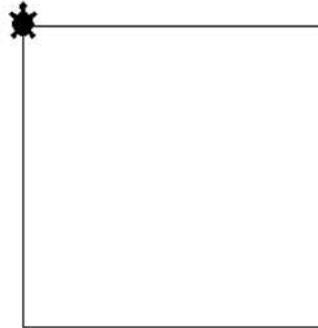
# 변수와 문자열을 연결하여 기사를 작성한다.
print("")
print("=====")
print(f"오늘 {stadium}에서 야구 경기가 열렸습니다.")
print(f"{winner}와 {loser}는 치열한 공방전을 펼쳤습니다.")
print(f"{vip}이 맹활약을 하였습니다.")
print(f"결국 {winner}가 {loser}를 {score}로 이겼습니다.")
print("=====")
```



## Lab: 사각형 그리기

- 사용자로부터 사각형의 크기를 입력받아서 크기에 맞는 사각형을 그려보자.

사각형의 크기는 얼마로 할까요? 200





# Sol: 사각형 그리기

```
##
#           이 프로그램은 사용자로부터 크기를 받아서 사각형을 그린다.
#
import turtle
t = turtle.Turtle()
t.shape("turtle")

# 사용자로부터 사각형의 크기를 받아서 size라는 변수에 저장한다.
# 사각형의 크기는 정수이므로 input()이 반환하는 문자열을 int()를 통하여 정수로 변환하였다.
size = int(input("사각형의 크기는 얼마로 할까요? "))

# 사각형을 다음과 같은 코드로 그린다. 이때 변수 size를 사용하자.
t.forward(size)      # size 만큼 거북이를 전진시킨다.
t.right(90)          # 거북이를 오른쪽으로 90도 회전시킨다.
t.forward(size)
t.right(90)
t.forward(size)
t.right(90)
t.forward(size)

turtle.done()
```



# Mini Project

- 여러 가지 입력으로 거북이를 조종하는 프로그램을 작성해보자.

색상(blue, red, yellow만 가능): blue

거리와 각도를 입력하시오.

거북이 회전 각도(left): 10

거북이 전진 거리: 200

제어에 성공하였습니다.

거리와 각도를 입력하시오.

거북이 회전 각도(left): 25

거북이 전진 거리: 100

제어에 성공하였습니다.

...





# Mini Project

- 이를 위하여 아직 학습하지 않았지만 다음과 같은 무한 반복 구문을 잠시 빌려서 사용하자.

```
while True:
```

```
    t.left(10)  
    t.forward(200)  
    ...
```

여기에 작성한 문장은 무한히 반복된다. 사용자는  
Ctrl+C를 눌러서 중지시킬 수 있다.



# 이번 장에서 배운 것

- 변수는 값을 저장하는 상자와 같은 것으로 저장된 값은 나중에 유용하게 사용될 수 있다.
- 변수들은 이름을 가지고 있다.
- 변수들은 숫자뿐만 아니라 문자열도 저장할 수 있다. 사실은 어떤 것이든지 저장 가능하다.
- 문자열은 큰따옴표("...")나 작은 따옴표('...')을 사용할 수 있다.
- `input()` 함수를 이용하여 사용자로부터 문자열을 받을 수 있다.
- 파이썬에서 기본 자료형은 `int`형, `float` 형, `str` 형이다.
- 문자열을 정수로 변경하려면 `int()` 함수를 사용한다.
- 문자열을 실수로 변경하려면 `float()` 함수를 사용한다.
- 정수나 실수를 문자열로 변경하려면 `str()`를 사용한다.





# Q & A

