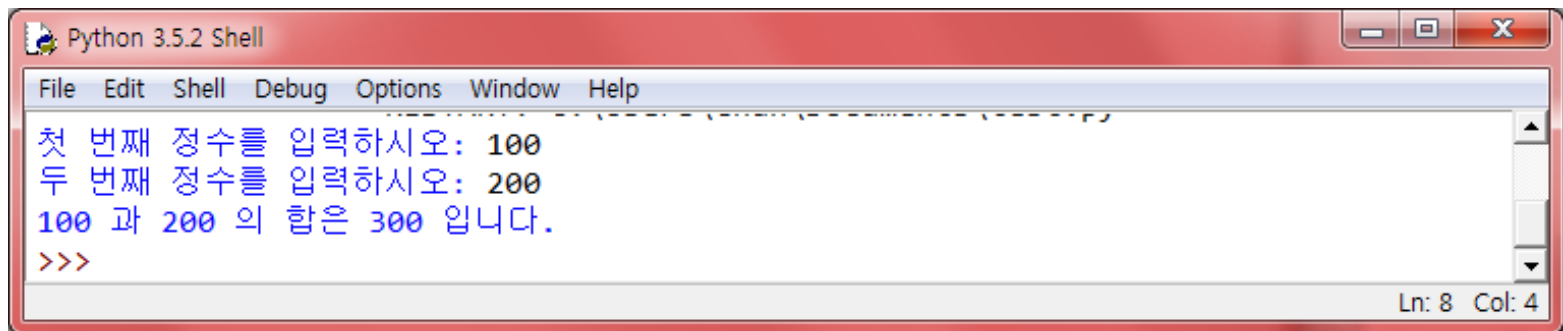




2장 변수를 소개합니다.

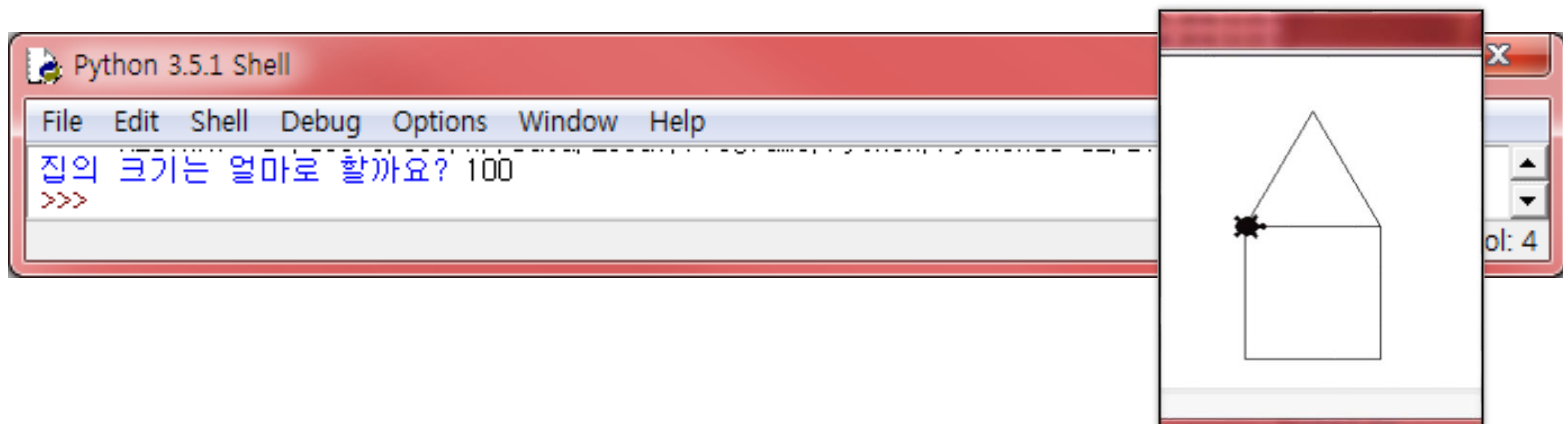
이번 장에서 학습하는 내용

- 변수(variables)를 사용하여 프로그램을 작성한다.
- 외부 입력을 받아 수행하는 프로그램을 작성한다.



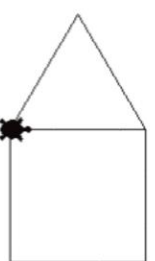
```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
첫 번째 정수를 입력하시오: 100
두 번째 정수를 입력하시오: 200
100 과 200 의 합은 300 입니다.
>>>
```

Ln: 8 Col: 4



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
집의 크기는 얼마로 할까요? 100
>>>
```

ol: 4



변수 (Variables)

변수

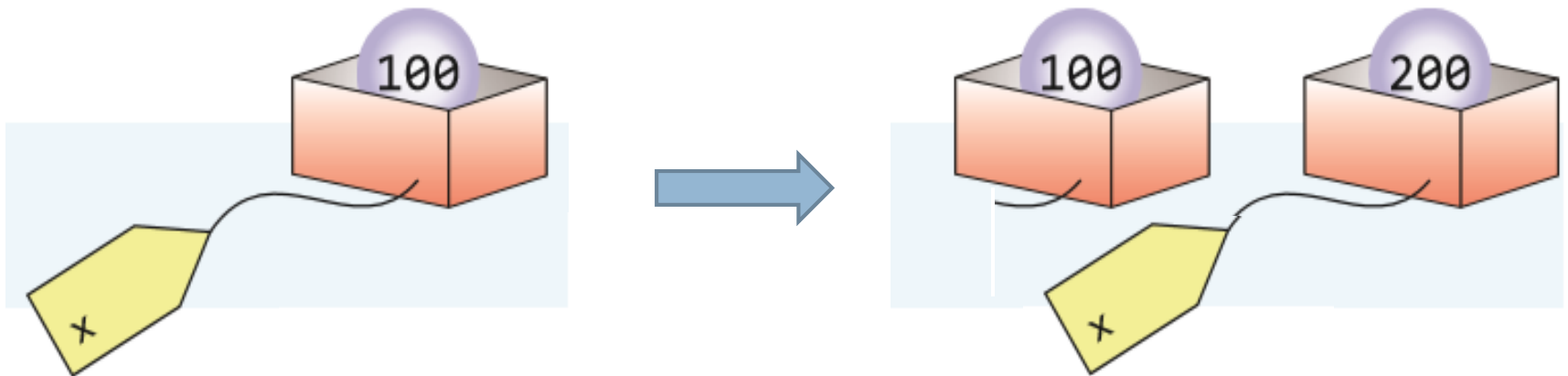
- 프로그램 안에서 값에 이름을 붙여 사용할 수 있도록 해준다.

```
>>> x = 100  
>>> pi = 3.141592  
>>> s = pi * x * x
```

변수의 사용

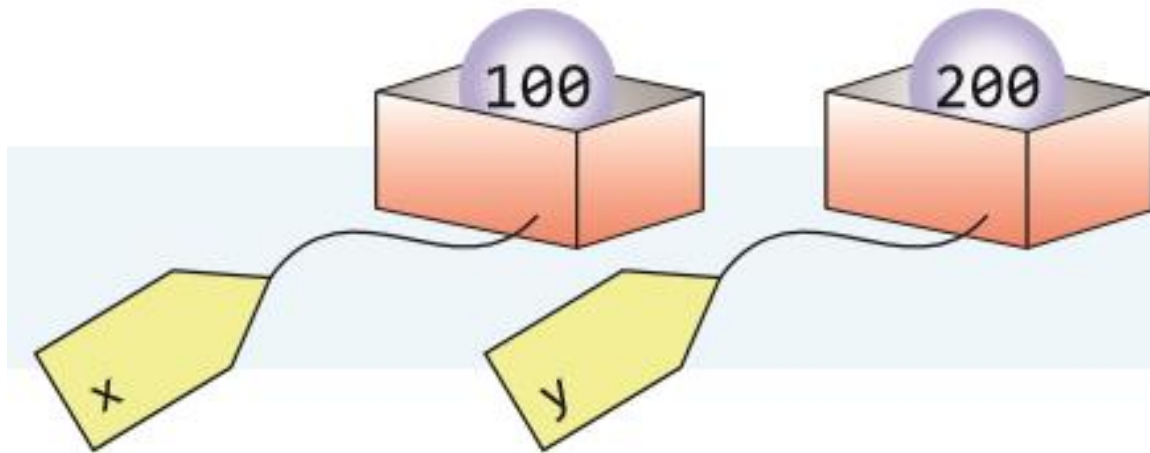
- 파이썬에서 변수에는 자유롭게 다른 값을 가리킬 수 있다.

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



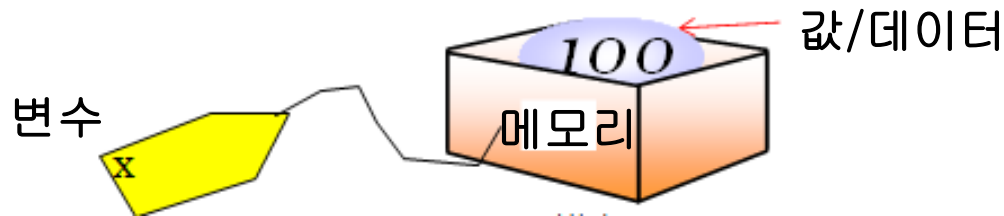
변수 2개 생성

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



변수의 소개

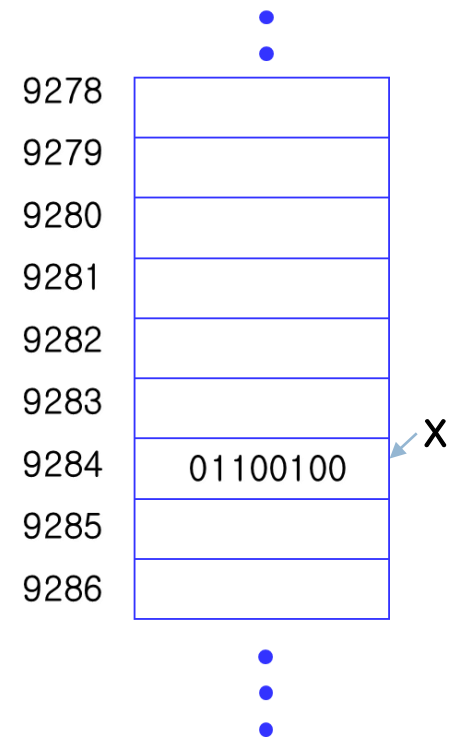
- 프로그래밍언어에서 변수(variable)는 값(value)을 저장하는 상자(메모리 장소)로 생각할 수 있다.
- 하지만, 파이썬에서는 변수는 메모리 상자 자체가 아닌 메모리 상자를 가리키는 이름으로 사용된다.
 - ▣ 이름은 바뀔 수도 있고, 다른 곳을 가리키는데 사용될 수도 있다.



x는 메모리 셀 9284번지이며 현재 100이 저장되어 있다.(C)

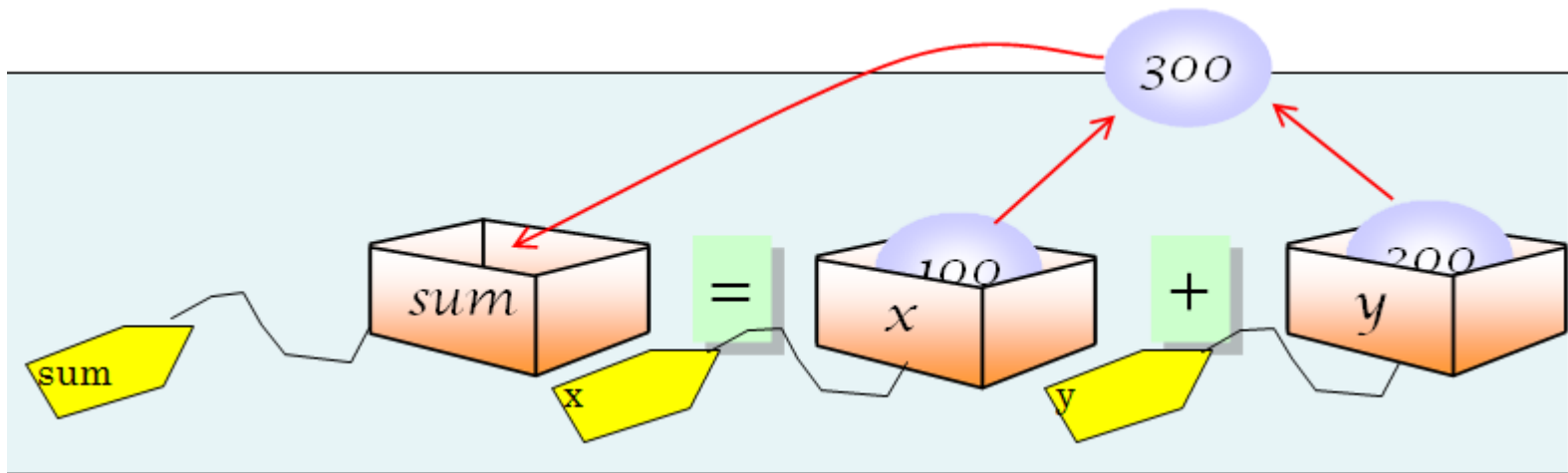
VS

x는 100을 저장한 메모리 셀 9284번지를 가리킨다.(Python)



변수를 이용한 계산

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



주의!!



경고

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.

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
>>> x = 2
>>> x == 3
False
>>> x
2
>>> |
```

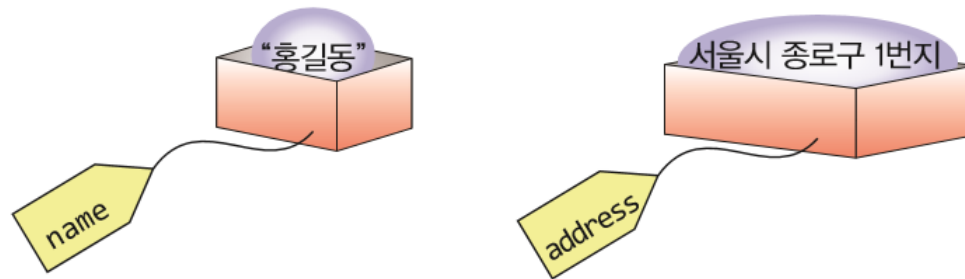
Ln: 16 Col: 4

수학의 =와 프로그래밍의 =는 다르다!

문자열도 변수에 저장할 수 있다!

- 파이썬의 변수에는 정수뿐만 아니라 문자열도 저장할 수 (가리킬 수) 있다.

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



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

도전문제



도전문제

무엇이 출력될까?

```
>>> x = 7
```

```
>>> y = 6
```

```
>>> print(x + y)
```

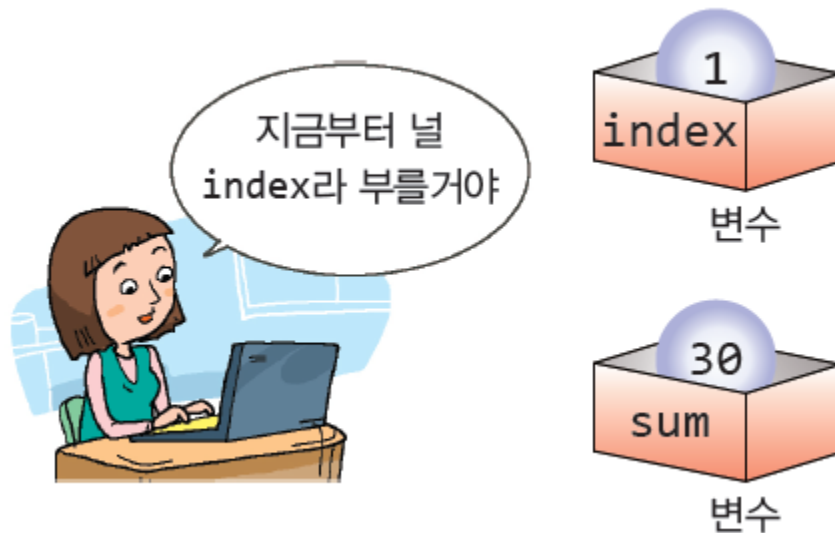
```
>>> x = '7'
```

```
>>> 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.



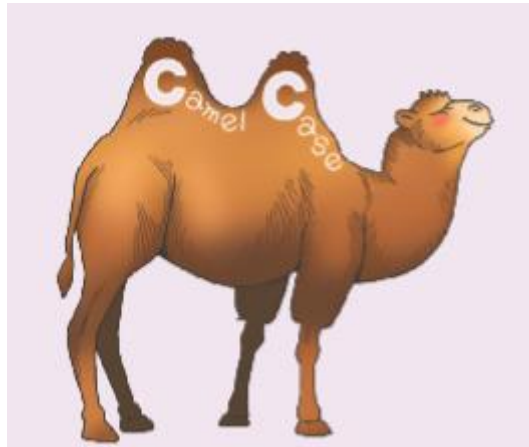
식별자(identifier, 이름) 규칙

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 (CamelCase)

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



변수도 수식(expression)의 일부이다!

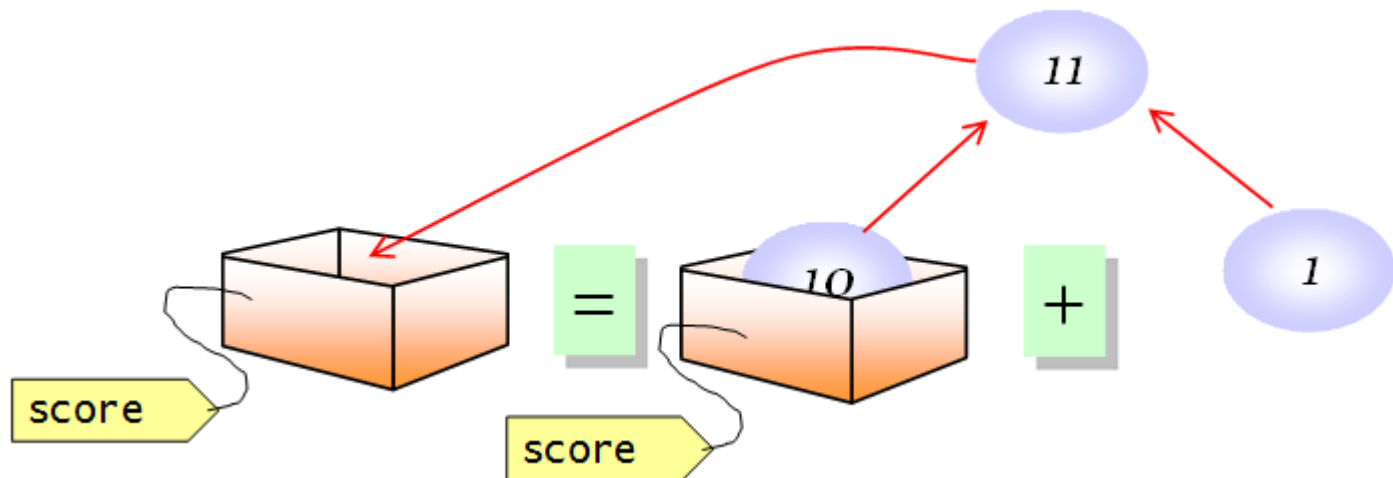
- 수식(계산식, expression): 계산하면 값이 되는 형태
 - ▣ 파이썬 셸은 수식을 받으면 값을 돌려준다.
- 파이썬 수식의 종류
 - ▣ 상수(literals)
 - 정수: 123, -20000
 - 부동 소수점 수 : 3.141592, 1.22e-2
 - 문자열: 'Hello World', "!", "12345"
 - ▣ 계산식 : 수식을 연산자로 연결
 - ▣ 변수
 - X, y, abc, hello
 - ▣ 함수식
 - math.sqrt(100)

지정문 (Assignment)

<변수> = <수식>

→ 오른쪽 수식을 계산하여 저장하고 이를 왼쪽 변수 이름으로 가리킨다.

```
score = 10  
score = score + 1
```



지정문 (Assignment)

<변수> = <수식>

→ 오른쪽 수식을 계산하여 저장하고 이를 왼쪽 변수 이름으로 가리킨다.

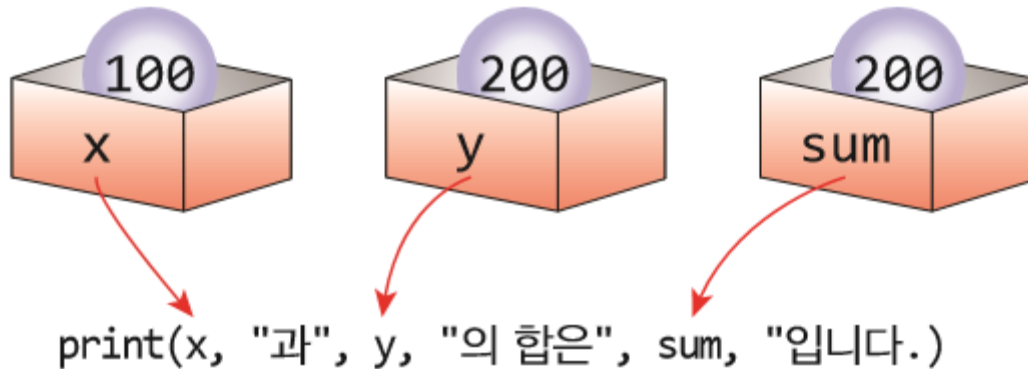
```
score = 10  
score = score + 1
```

- “=” 양쪽에서 변수의 의미 차이
 - ▣ 왼쪽 (L-value) : 값을 저장한 상자를 가리키는 이름
 - ▣ 오른쪽 (R-value) : 변수가 가리키는 값

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.

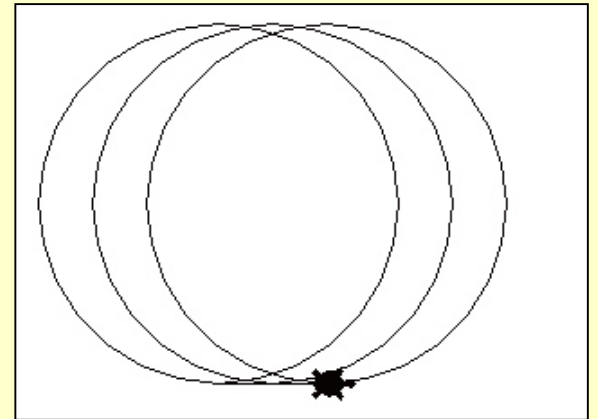


변수는 어디에 유용할까?



- 다음과 같이 터틀 그래픽을 사용하여 반지름이 100픽셀인 3개의 원을 그리는 프로그램이 있다고 하자.
- 100이 중복되어 사용되고 있다.

```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
  
t.circle(100) # 반지름이 100인 원이 그려 진다.  
t.fd(30)  
t.circle(100) # 반지름이 100인 원이 그려 진다.  
t.fd(30)  
t.circle(100) # 반지름이 100인 원이 그려 진다.
```



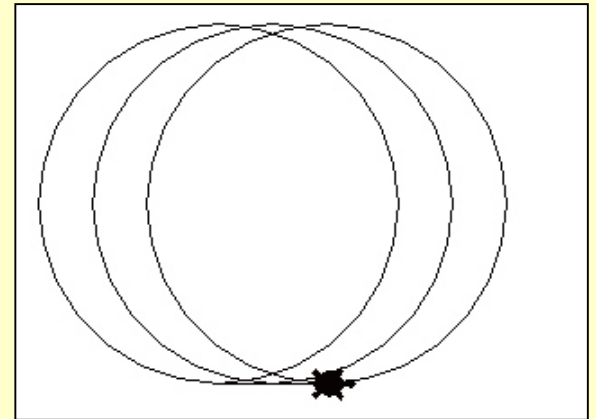
변수는 어디에 유용할까?



- 값의 중복을 피할 수 있다!

```
import turtle
t = turtle.Turtle()
t.shape("turtle")

radius = 100
t.circle(radius) # 반지름이 100인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 100인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 100인 원이 그려 진다.
```



변수는 어디에 유용할까?



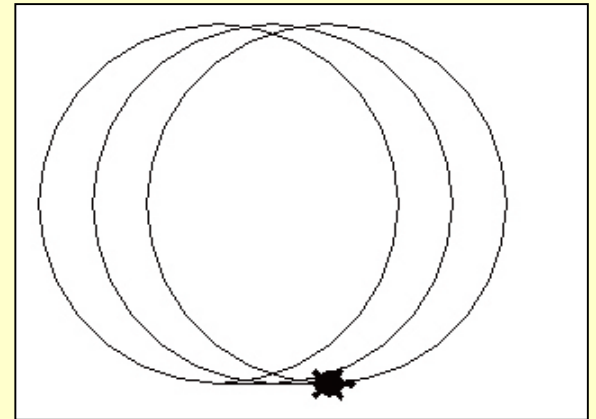
- 값의 중복을 피할 수 있다!

→ 수정이 용이하다!

반지름을 60으로 바꾸려면 ...

```
import turtle
t = turtle.Turtle()
t.shape("turtle")

radius = 60
t.circle(radius) # 반지름이 100인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 100인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 100인 원이 그려 진다.
```



변수는 어디에 유용할까?



- 계산의 중복도 피할 수 있다!

반지름을 루트 120으로 하려면...

```
import turtle
Import math
t = turtle.Turtle()
t.shape("turtle")

t.circle(math.sqrt(120)) # 반지름이 루트 120인 원이 그려 진다.
t.fd(30)
t.circle(math.sqrt(120)) # 반지름이 루트 120인 원이 그려 진다.
t.fd(30)
t.circle(math.sqrt(120)) # 반지름이 루트 120인 원이 그려 진다.
```

변수는 어디에 유용할까?



- 계산의 중복도 피할 수 있다!

반지름을 루트 120으로 하려면...

```
import turtle
Import math
t = turtle.Turtle()
t.shape("turtle")

radius = math.sqrt(120)
t.circle(radius) # 반지름이 루트 120인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 루트 120인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 루트 120인 원이 그려 진다.
```

변수는 어디에 유용할까?



- 복잡한 계산을 나누어서 간단하게 표현한다.

```
원금 = 10000.0
```

```
월이자율 = 0.01
```

```
개월 = 12
```

```
결산액 = 원금 + 원금*월이자율*개월
```

```
print(결산액)
```

```
원금 = 10000.0
```

```
월이자율 = 0.01
```

```
개월 = 12
```

```
이자 = 원금*월이자율*개월
```

```
결산액 = 원금 + 이자
```

```
print(결산액)
```


변수의 유용성

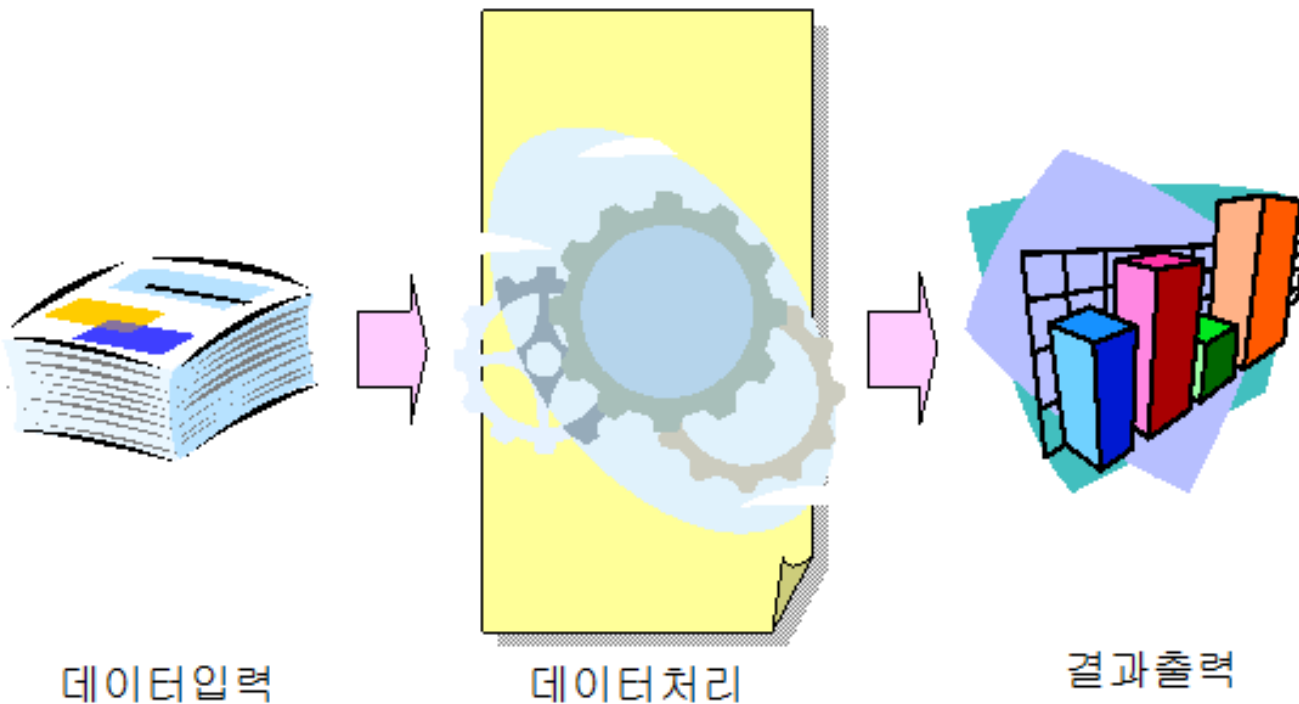
- 값 또는 식의 중복을 피할 수 있다.
 - ▣ 프로그램 수정 용이
 - ▣ 컴퓨터 계산량을 줄일 수 있다.
- 복잡한 식을 나누어서 간단하게 표현
- 미리 알 수 없는 값(입력값)에 이름을 붙일 수 있다.



입력 (Input)

일반적인 프로그램의 구조

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



입력 (Input)

- 입력이 없다면 ?
 - ▣ 입력이 없다면 항상 같은 동작만을 반복할 것이다.

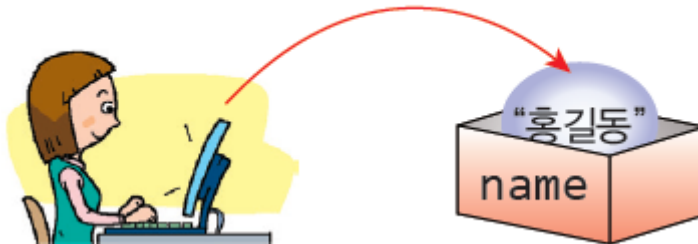


사용자로부터 입력받기

- input 함수를 사용한다.
 - ▣ 키보드 입력을 기다리다가 문자열을 만들어 준다.
 - ▣ 기다릴 때 출력할 메시지를 문자열 인자로 받는다.

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

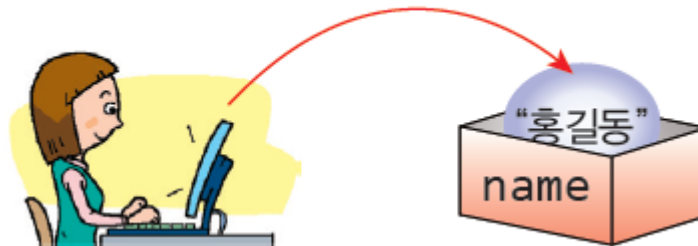
이름을 입력하시오: 홍길동
홍길동



사용자로부터 입력받기

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

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



사용자로부터 정수값 입력받기

- Input 함수는 사용자의 키보드 입력을 문자열로 간주한다.

```
num = input( " 숫자를 입력하시오: ")  
print(num + 100)
```

숫자를 입력하시오: 123

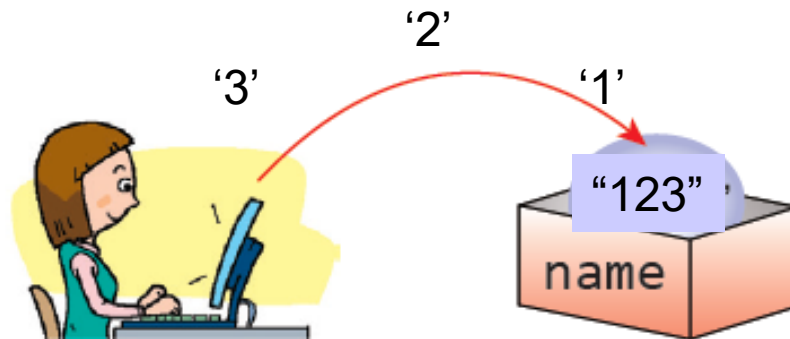
Traceback (most recent call last):

File "<pyshell#1>", line 1, in <module>

print(num + 100)

TypeError: can only concatenate str (not "int") to str

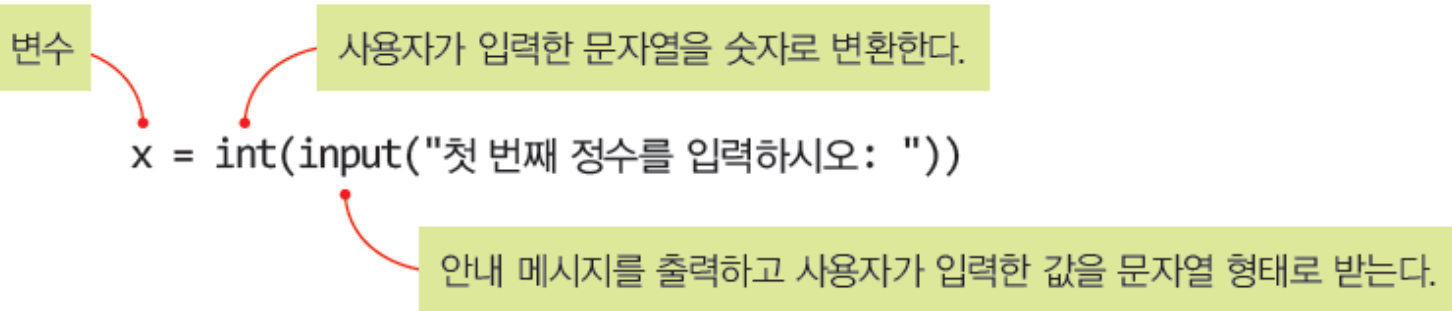
>>>



사용자로부터 정수값 입력받기

- 정수의 입력
 - ▣ int 함수를 사용하여 문자열을 정수로 변경할 수 있다!
 - ▣ int는 "정수 모양의" 문자열을 정수로 바꾸어준다.

input() 사용법



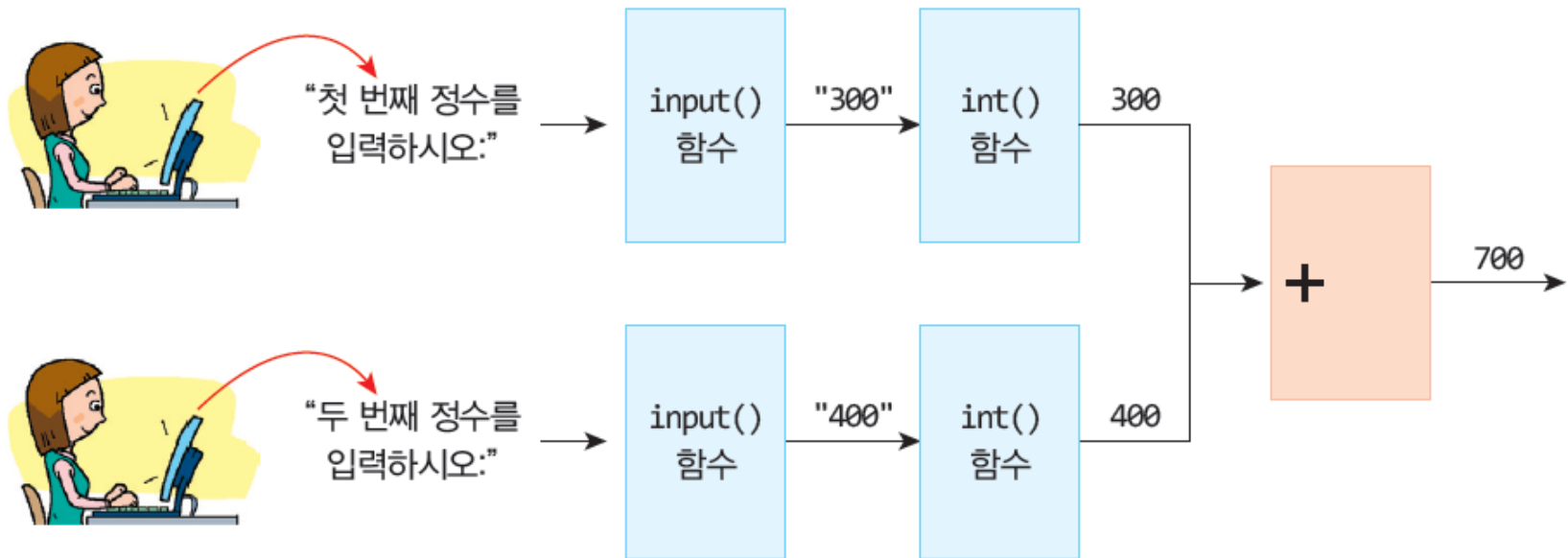
- 함수는 합성(composition)하여 사용할 수 있다.

정수를 입력받는 프로그램

```
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.

함수의 합성



도전문제



도전문제

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

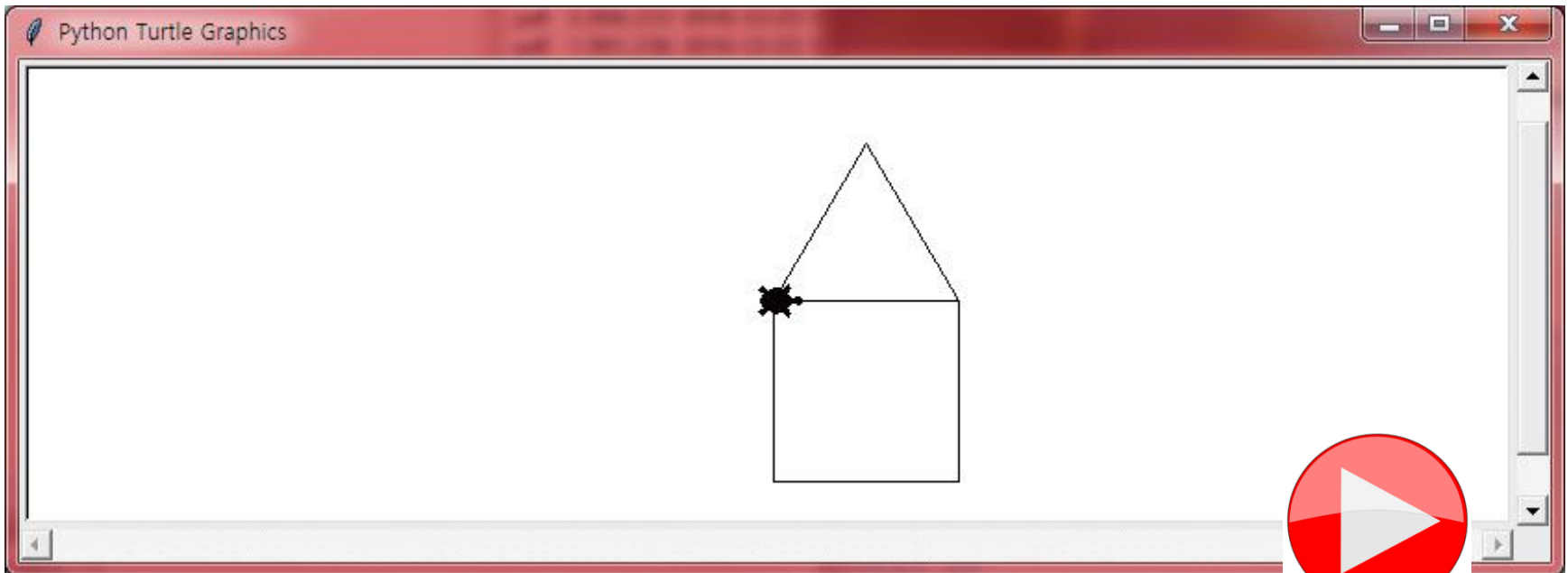
```
이름을 입력하시오: 홍길동
홍길동 씨, 안녕하세요?
파이썬에 오신 것을 환영합니다.
첫 번째 정수를 입력하시오: 300
두 번째 정수를 입력하시오: 400
300 과 400 의 합은 700 입니다.
```

Lab: 집그리기



- 우리는 사용자로부터 집의 크기를 입력받아서 크기에 맞는 집을 그려보자.

집의 크기는 얼마로 할까요? 100



Run Python

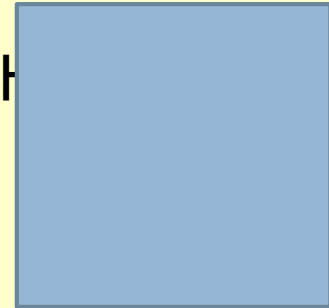
```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")
```

```
size = int(input("집의 크기는 얼마로 할까요? "))
```

```
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)
```

```
t.right(90)
```

```
t.forward(size)  
t.left(120)  
t.forward(size)  
t.left(120)  
t.forward(size)  
t.left(120)
```

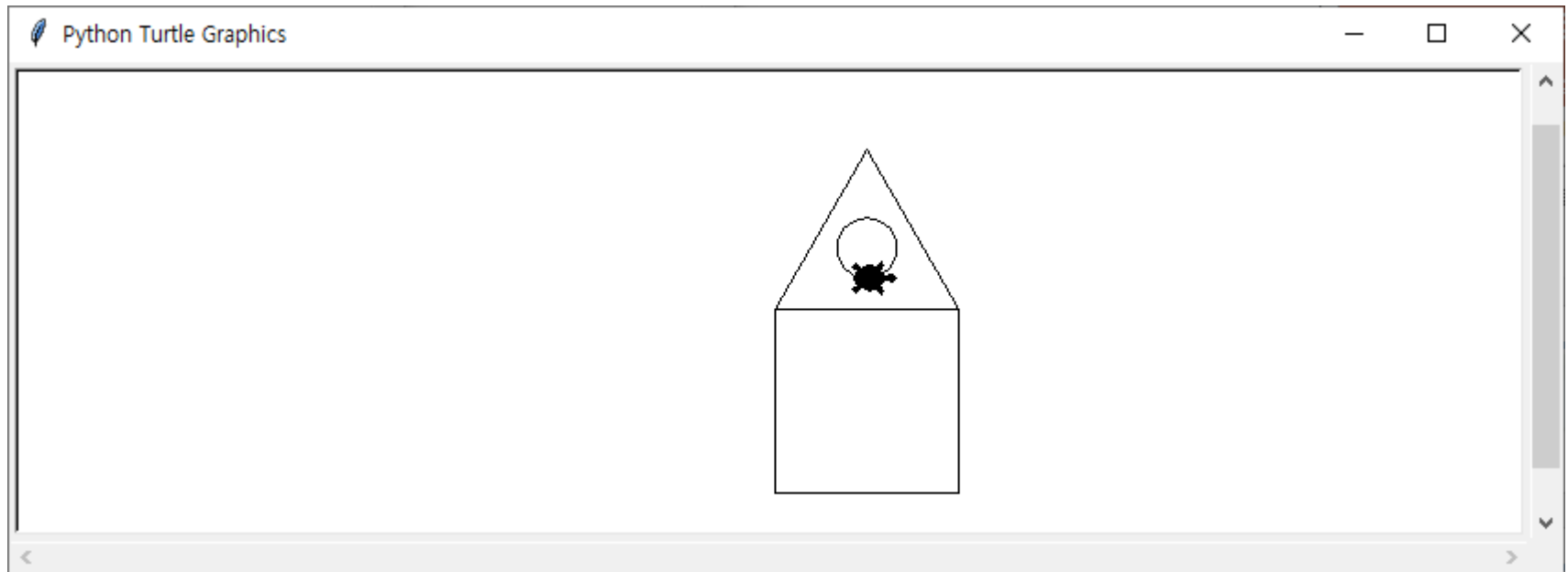


Lab: 집에 창 추가하기



- 그리지 않고 이동하도록 펜을 들었다(up) 놓아야(down) 한다.

집의 크기는 얼마로 할까요? 100



S
import turtle
t = turtle.Turtle()
t.shape("turtle")

size = int(input("집의 크기는 얼마로 할까요? "))

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)

t.right(90)

t.forward(size)
t.left(120)
t.forward(size)
t.left(120)
t.forward(size)
t.left(120)

t.up()
t.forward(size/2)
t.left(90)
t.forward(size/6)
t.right(90)
t.down()
t.circle(size/6)

Lab: 로봇 기자 만들기



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

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

=====

오늘 서울 에서 야구 경기가 열렸습니다.
삼성 과 **LG** 은 치열한 공방전을 펼쳤습니다.
홍길동 이 맹활약을 하였습니다.
결국 삼성 가 **LG** 를 8:7 로 이겼습니다.

=====



Solution

```
# 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

- 프로그램에서 변수를 사용하여 값을 가리킬 수 있다.
- 변수들은 숫자뿐만 아니라 문자열도 가리킬 수 있다.
- 변수들의 이름은 식별자 규칙에 따라 자유롭게 만들 수 있다.
- 변수를 사용하여 이해하기 쉽고 관리가 용이한 프로그램을 만들 수 있다.
- 입력을 통해 주어진 값을 사용하여 동작하는 프로그램을 작성할 수 있다.
- 수학에서처럼 파이썬에서도 함수를 합성하여 사용할 수 있다.

