

## 2. 변수

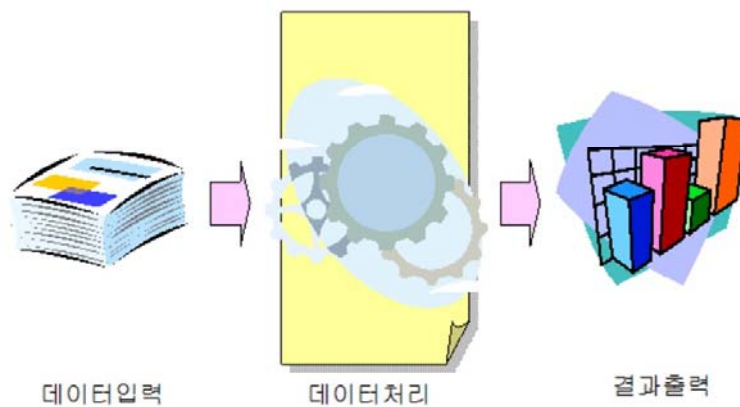
### 학습 내용

- 변수가 무엇인지 이해합니다.
- 입력과 출력의 방법을 살펴봅니다.
- 간단한 계산기 프로그램을 작성해봅니다.
- 간단한 퀴즈 프로그램을 작성해봅니다.
- `print()` 함수의 사용법을 자세히 살펴봅니다.

## 일반적인 프로그램의 구조

### ●일반적인 프로그램

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



## 이번 장에서 학습하는 내용

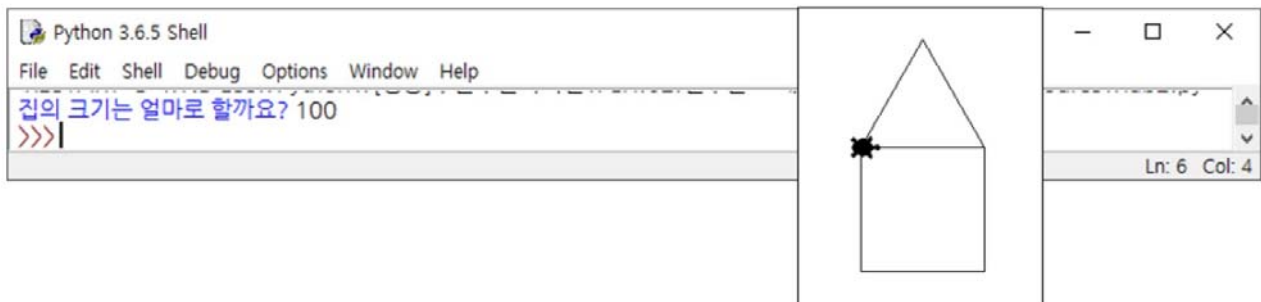
(1) 첫 번째 프로그램은 두 개의 정수를 가지고 덧셈 연산을 실행한 후에, 연산 결과를 화면에 출력한다.



```
Python 3.6.5 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: 11 Col: 4

(2) 사용자로부터 집의 크기를 입력받아서 터틀 그래픽으로 화면에 그린다.



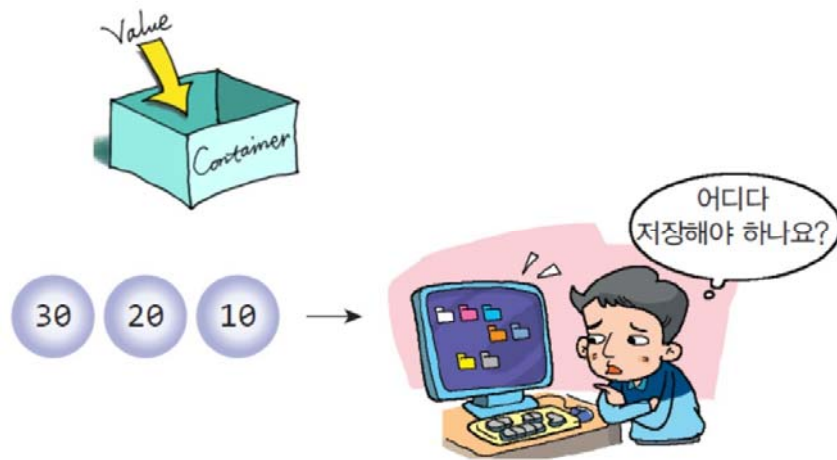
## 주의 사항

- 이 단원부터 다루는 모든 실습문제중에서 파일 이름이 지정되어 있는 경우에는 파일로 작성하여야 합니다.
- 작성한 파일을 실행하고 결과와 소스코드를 과제로 제출하세요.
  - 결과는 화면 캡처를 사용하세요.
  - 파일로 저장하지 않은 경우에는 IDLE에서 실행한 결과를 화면 캡처하면 됩니다.
- 결과 파일명은 다음과 같이 정하세요.
  - 소스코드가 A.py 일 경우 → A.jpg 또는 A.png
- 실습한 모든 내용은 하나의 압축파일로 작성하세요.
  - 압축 파일명 : 실습20190308-학번-이름.zip
- 실습시간 종료 전에 LMS에 있는 해당 과제에 제출하세요.
  - 실습 시간 이후에는 제출할 수 없습니다.

## 변수의 소개

### ●변수(variable)

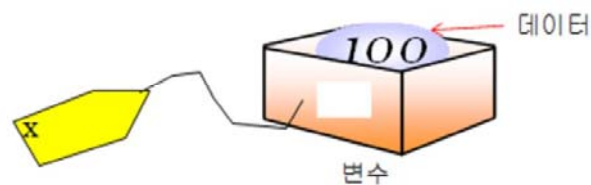
- 값을 저장하는 공간
- 값을 저장하는 상자로 생각할 수 있다.



## 변수 생성

- 파이썬에서 변수를 생성하려면 다음과 같이 한다.

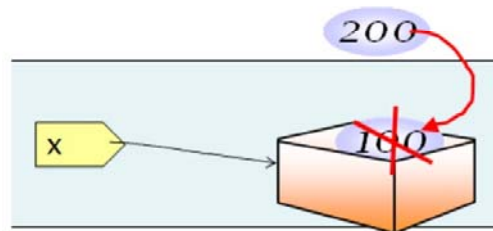
```
>>> x = 100
>>>
```



## 변수의 사용

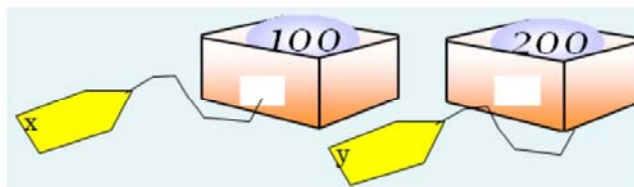
- 생성된 변수에는 얼마든지 다른 값을 저장할 수 있다.

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



## 변수 2개 생성

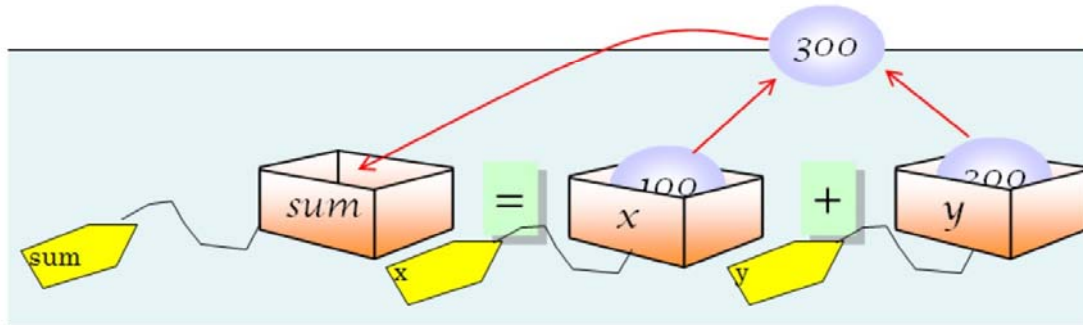
```
>>> x = 100  
>>> y = 200
```



## 변수를 이용한 계산 (실습 - P0201)

●IDLE에서 실행하세요. (캡처 화일명 : P01.jpg 나 P01.png)

```
>>> 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.  
파이썬 = 기호는 “변수에 값을 저장하라”라는 의미이다  
혼동하지 않도록 하자 등호는 == 와 같이 표시한다

## 변수는 문자열도 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 (실습 - P0202)

- 무엇이 출력될까?

- IDLE에서 실행하세요. (캡처 화일명 : P02.jpg 나 P02.png)

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

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



## 변수의 이름

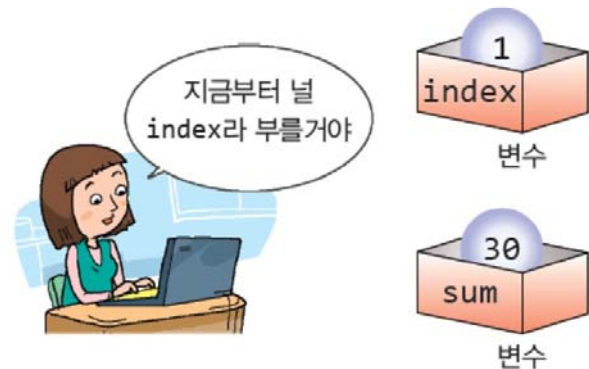
### ●식별자(Identifier)

- 식별하기 위하여 사용하는 이름

### ●식별자 만드는 규칙

- 의미 있는 이름을 사용한다.
- 식별자는 **영문자와 숫자, 밑줄(\_)**로 이루어진다.
- 변수의 이름 중간에 공백이 들어가면 안 된다. 단어를 구분하려면 밑줄(\_)을 사용 한다.
- 첫 글자는 반드시 영문자 또는 밑줄 문자( \_ )이어야 한다. (숫자로 시작할 수 없다)
- 소문자와 대문자는 서로 다르게 취급된다.
  - ◆index, Index, INDEX 모두 다르게 취급된다.

### ●변수 이름은 식별자의 일종이다.



## 식별자

### ●유효한 식별자

|                    |                           |
|--------------------|---------------------------|
| 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. |

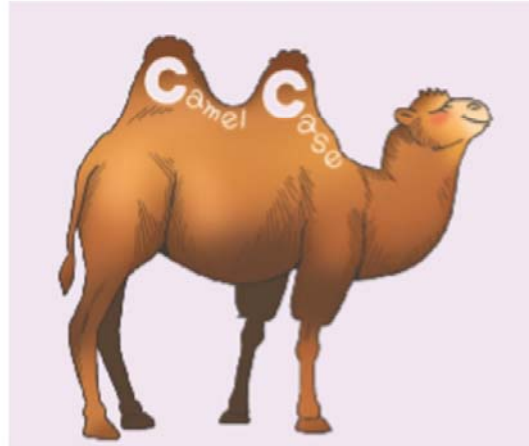
### ●유효하지 않은 식별자

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

## 낙타체

### ● 낙타체

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



## 파이썬 키워드

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



## 할당문

### ● 할당문 Assignment Statement

- 어떤 값을 1개의 변수에 할당하는 명령문

### ● 할당 연산자 Assignment Operator

- 할당문에서 사용하는 연산자 (=)

|   |                         |
|---|-------------------------|
| <code>y = 1</code>                            | # 1을 변수 y에 할당한다.        |
| <code>radius = 1.0</code>                     | # 1.0을 변수 radius에 할당한다. |
| <code>x = 5 * ( 3 / 2 ) + 3 * 2</code>        | # 표현식의 결과를 x에 할당한다.     |
| <code>x = y + 1</code>                        | # y와 1의 합을 x에 할당한다.     |
| <code>area = radius * radius * 3.14159</code> | # 넓이(area)를 계산한다.       |

```
i = j = k = 1
```

```
k = 1  
j = k  
i = j
```

모든 변수는 사용할 수 있는 영역(스코프)을 가지고 있다.

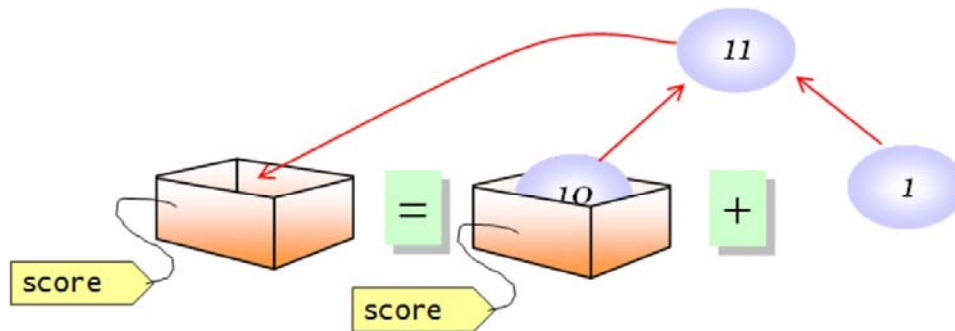
count가 아직 정의되지 않음.

```
>>> count = count + 1  
Name Error: count is not defined  
>>>
```

```
>>> count = 1          # count가 생성됨.  
>>> count = count + 1  # 이제 count를 증가시킴.  
>>>
```

## 이런 것도 가능하다!

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

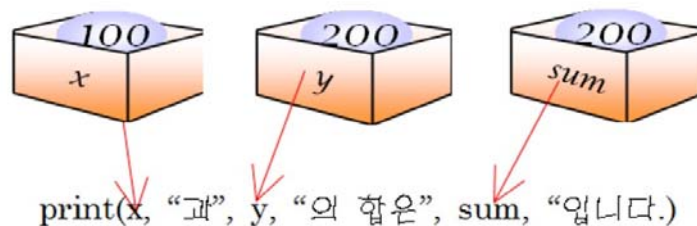


## 여러 값을 함께 출력하기 (실습 - P0203)

●파일명 : P03.py

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

100 과 200 의 합은 300 입니다.



```
var1, var2, ... , varn = exp1, exp2, ... , expn
```

- 오른쪽에 있는 모든 표현식들을 계산한다.
- 계산된 값을 왼쪽 부분에 각각 대응하는 변수로 동시에 할당한다.

### ● 변수값 교환하기

```
>>> x = 1
>>> y = 2
>>> temp = x      # x를 임시 변수 temp에 저장한다.
>>> x = y          # y의 값을 x에 할당한다.
>>> y = temp       # temp의 값을 y에 할당한다.
```

```
>>> x, y = y, x    # x와 y를 교환한다.
```

## 상수

### ● 상수 Constant

- 절대로 바뀌지 않는 영구 데이터

◆ 예 :  $\pi = 3.14159$

- 상수를 나타내기 위해 사용하는 변수 이름은 대문자만 사용한다.

◆ 일반 변수와 구분하기 좋다.

- 상수 변수를 사용할 때의 이점

◆ 동일한 값이 여러 번 사용될 때, 그 값을 반복해서 타이핑하지 않아도 된다.

◆ 상수 값을 변경해야 한다면(예를 들어, PI를 3.14에서 3.14159로 변경), 소스 코드의 한 위치에 한 번만 변경하면 된다.

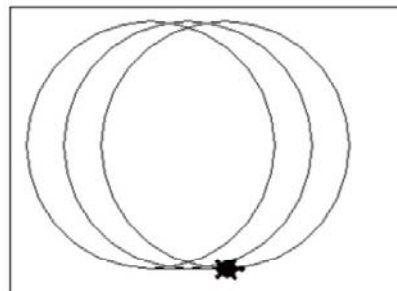
◆ 서술 이름을 사용하므로 프로그램을 읽기 쉽다.

## ●상수 Constant

```
# 반지름을 할당한다.  
radius = 20 # 반지름은 이제 20이 된다.  
  
# 넓이를 계산한다.  
PI = 3.14159  
area = radius * radius * PI  
  
# 결과를 출력한다.  
print("반지름이", radius, "인 원의 넓이는", area, "입니다.")
```

## Lab-1: 변수는 어디에 유용할까? (실습 - P0204)

- 다음과 같이 터틀 그래픽을 사용하여 반지름이 100픽셀인 3개의 원을 그리는 프로그램을 작성하시오.



- turtle.shape("turtle")
  - ◆거북이의 모양이 나타난다.
- 원을 그리는 명령어는 circle(반지름) 이다.



## Lab-1: 변수는 어디에 유용할까? (실습 - P0204)

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

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

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

## 코드 변경 (실습)

- 하지만 갑자기 원의 반지름을 50으로 변경하여서 다시 그려야 한다면 어떨까?
- 원의 반지름이 변수로 표현되었기 때문에 쉬운 방법이 있다. 한군데만 고치면 된다.

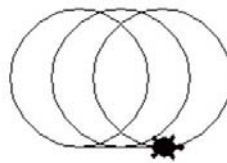
|   |   |
|---|---|
| <pre>t.circle(<del>100</del>) t.fd(30) t.circle(<del>100</del>) t.fd(30) t.circle(<del>100</del>)</pre> <p>50</p> | <pre>radius = <del>100</del> t.circle(radius) t.fd(30) t.circle(radius) t.fd(30) t.circle(radius)</pre> <p>50</p> |
|---|---|



## Solution (실습)

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

radius = 50
t.circle(radius) # 반지름이 50인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 50인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 50인 원이 그려 진다.
```



## 사용자로부터 정수 입력받기

### input() 사용법

변수

사용자가 입력한 문자열을 숫자로 변환한다.

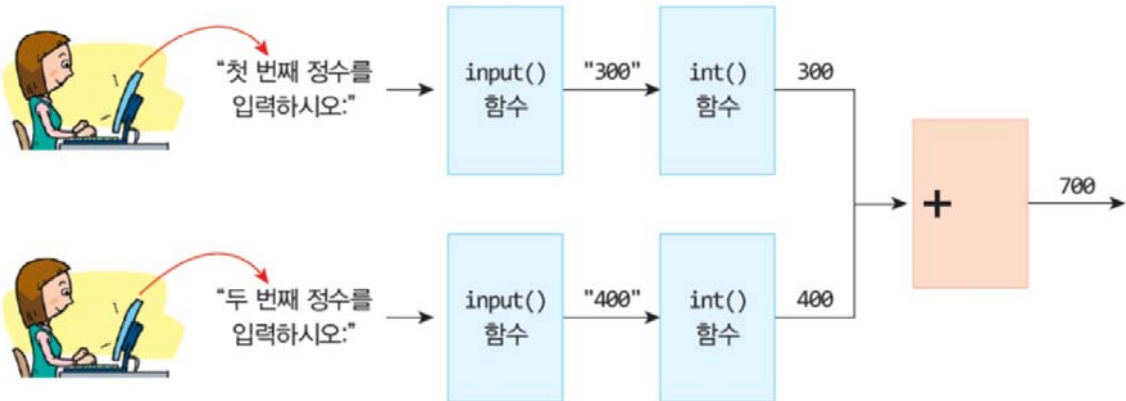
```
x = int(input("첫 번째 정수를 입력하시오: "))
```

안내 메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 받는다.

## 완전한 코드 (실습 - P0205)

```
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.")
```

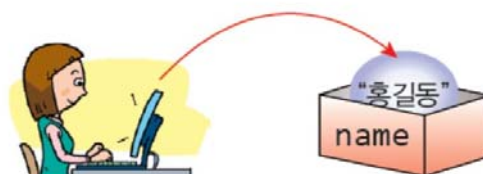
첫 번째 정수를 입력하시오: 300  
두 번째 정수를 입력하시오: 400  
100 과 200 의 합은 300 입니다.



## 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 (실습 - P0206)



### 도전문제

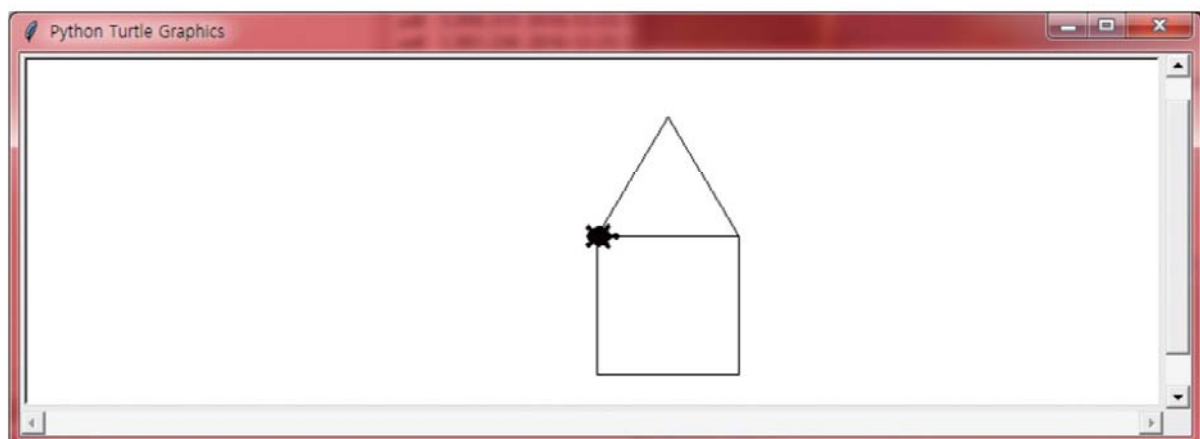
사용자의 이름을 물어보고 이어서 2개의 정수를 받아서 덧셈을 한 후에 결과를 출력하는 다음과 같은 프로그램을 작성해보자.

이름을 입력하시오: 홍길동  
홍길동 씨, 안녕하세요?  
파이썬에 오신 것을 환영합니다.  
첫 번째 정수를 입력하시오: 300  
두 번째 정수를 입력하시오: 400  
100 과 200 의 합은 300 입니다.

## Lab: 집그리기 (실습 - P0207)

- 우리는 사용자로부터 집의 크기를 입력받아서 크기에 맞는 집을 그려보자.

집의 크기는 얼마로 할까요? 100



## Solution

```
# 터틀 그래픽을 사용하려면 하므로 다음과 같은 코드를 소스 파일에 입력한다.
import turtle
t = turtle.Turtle()
t.shape("turtle")

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

# 집을 그릴 차례이다. 사각형을 다음과 같은 코드로 그린다. 이때 변수 size를 사용하자.
# 사각형을 그린다.
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)

# 이제 지붕을 그릴 차례이다. 현재 거북이는 위를 보고 있기 때문에
# 거북이를 오른쪽으로 90도 회전시켜서 오른쪽을 보도록 한다.
t.right(90)

# 지붕을 그리면 된다. 지붕은 간단히 삼각형으로 그렸다.
t.forward(size)
t.left(120)
t.forward(size)
t.left(120)
t.forward(size)
t.left(120)
```



2019-03-14

© Chang Seung Kim - All rights reserved

33

## Lab: 로봇 기사 만들기 (실습 - P0208)

- 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.
삼성 과 LG 은 치열한 공방전을 펼쳤습니다.
홍길동 이 맹활약을 하였습니다.
결국 삼성 가 LG 를 8:7 로 이겼습니다.
=====
```

2019-03-14

© Chang Seung Kim - All rights reserved

34



## Lab: 로봇 기자 만들기 (실습 - P0208)

```
# 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("=====")
```

## 동시 할당문 (실습 - P0209)

- 사용자로부터 3개의 숫자를 입력받고, 이들의 평균을 계산한다.
- `ComputeAverageWithSimultaneousAssignment.py`

```
1 # 사용자로부터 세 개의 숫자를 입력받는다.
2 number1, number2, number3 = eval(input(
3     "세 개의 숫자를 콤마(,)로 구분하여 입력하세요: ")
4
5 # 평균을 계산한다.
6 average = (number1 + number2 + number3) / 3
7
8 # 결과를 출력한다.
9 print(number1, number2, number3,
10       "의 평균은", average, "입니다.")
```

세 개의 숫자를 콤마(,)로 구분하여 입력하세요: 3,6,9  
3 6 9 의 평균은 6.0 입니다.



## 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.