

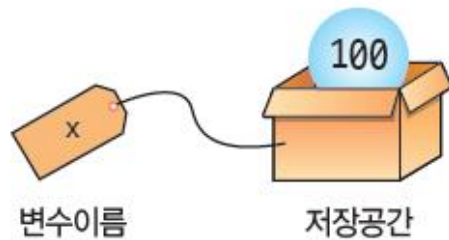
## 2장 변수와 자료형





# 변수. p45

- 변수(variable)
  - 값을 저장하는 메모리 공간에 이름을 붙이는 것
  - 값을 저장하는 상자로 생각하면 이해하기 쉽다.



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



# 변수 정의하기. p45

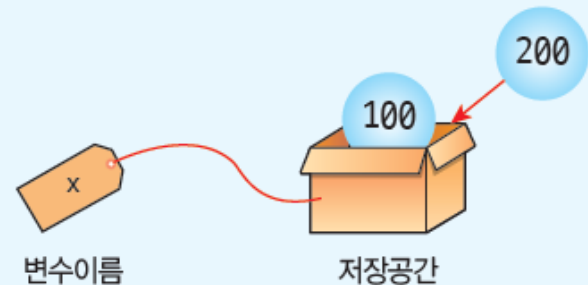
- 파이썬에서는 변수에 값을 저장하면 변수가 자동으로 생성된다.(다른 언어에서는 미리 변수를 선언하는 절차가 필요하다.)

변수 = 값

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

- 변수에 저장된 값은 변경할 수 있다.

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



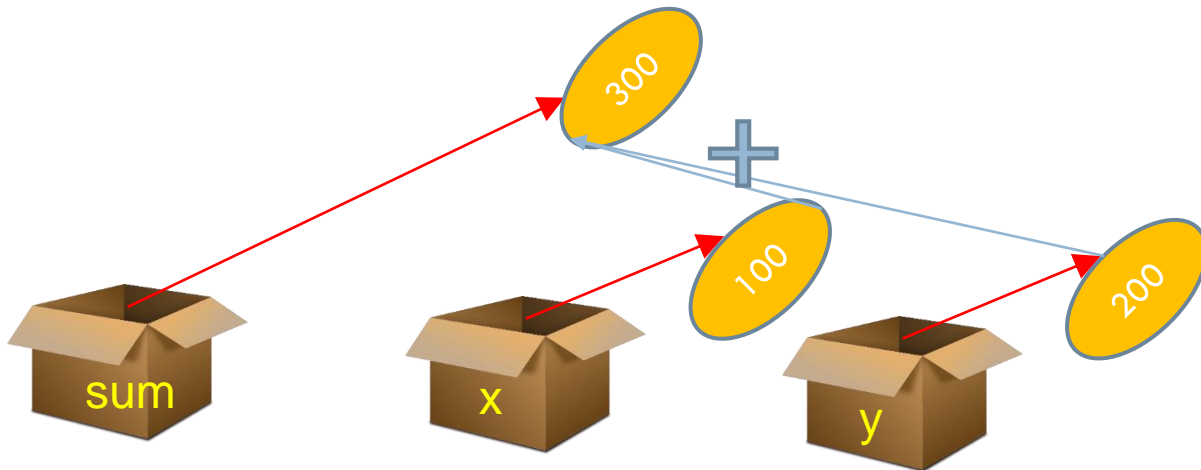


# Example 두 수의 합 계산하기. p46

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

# 변수 **x**를 생성하고 100을 저장한다.  
# 변수 **y**를 생성하고 200을 저장한다.  
# 변수 **sum**을 생성하고 **x+y**를 저장한다.

합은 300



- 수학에서의 변수와의 차이점

프로그래밍의 변수는 긴 이름을 상대적으로 사용을 설명하는 반면, 수학의 변수는 종종 간결함을 위해 1글자 또는 2글자 길이의 이름을 갖는다.

- 경고

파이썬에서 = 기호는 '변수에 값을 저장하라' 라는 의미이다. 혼동하지 않도록 하자. 등호는 파이썬에서 == 와같이 표시한다.

- 중간점검

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



# 변수의 이름. p47

- 의미 있는 이름을 사용하는 것이 좋다.
- 규칙
  - 첫 글자는 영문자 또는 밑줄(\_) 문자로 시작해야 하며 나머지 문자는 문자, 숫자 또는 밑줄이어야 한다.
  - #과 같은 기호는 사용할 수 없다. 이름 안에 공백도 허용되지 않는다.
  - 소문자와 대문자는 서로 다르게 취급된다.
  - if와 같은 예약어를 이름으로 사용할 수 없다. 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. p48

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.



# Example 원의 면적 계산하기. p48

반지름 20인 원의 면적= 1256.0

## 알고리즘

STEP #1. 사용자로부터 원의 반지름을 입력받는다.

STEP #2. 공식을 적용하여 면적을 계산한다.

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

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)
```



- 경고

변수의 이름을 정할 때 ....

- 참고사항 : 긴 변수 이름

변수 이름이 여러 개의 단어로 이루어진 ...

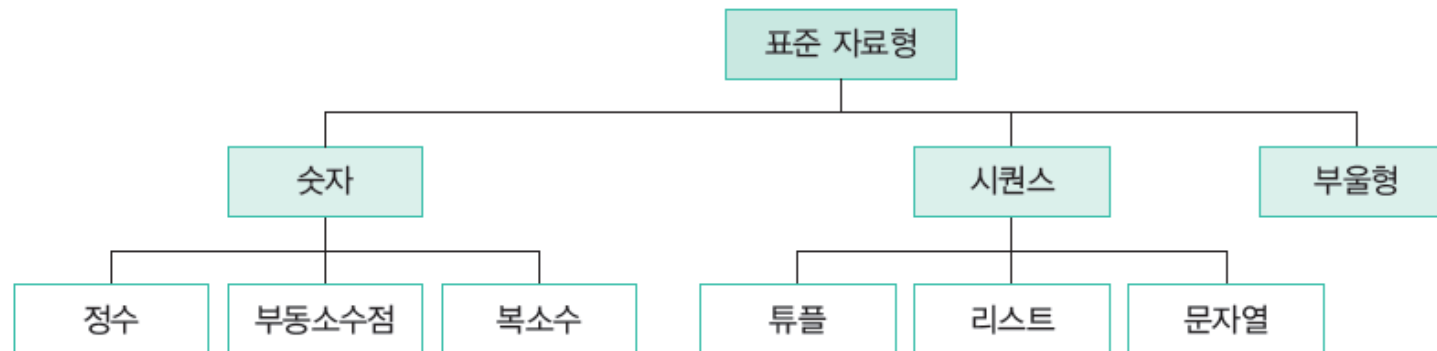
- 중간점검

1. 변수 이름의 ...
2. 파이썬에서 ...
3. 파이썬에서는 ...



# 자료형. p50

- 변수가 저장할 수 있는 데이터의 종류



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



# 자료형을 알려면. p51

```
>>> 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'>
```



- [illegible]

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

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

255 63 15

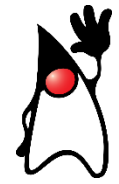
| Dec | Hex | Oct | Bin  |
|-----|-----|-----|------|
| 0   | 0   | 000 | 0000 |
| 1   | 1   | 001 | 0001 |
| 2   | 2   | 002 | 0010 |
| 3   | 3   | 003 | 0011 |
| 4   | 4   | 004 | 0100 |
| 5   | 5   | 005 | 0101 |
| 6   | 6   | 006 | 0110 |
| 7   | 7   | 007 | 0111 |
| 8   | 8   | 010 | 1000 |
| 9   | 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 |



# 진법변환은 할 줄 알아야 !

## 10진법, 2진법, 8진법, 16진법

| 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. p52

- 실수를 나타내는 자료형이다.

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

```
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
```

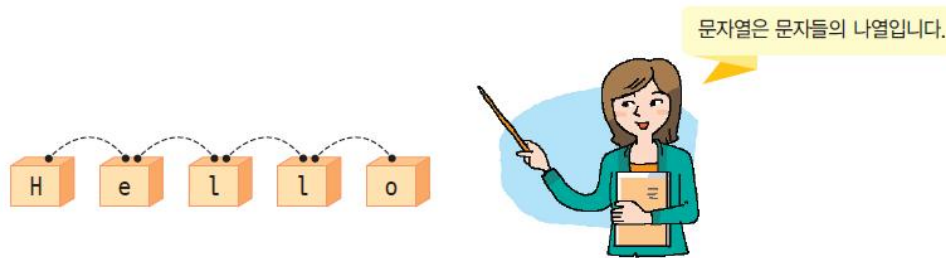


- 부동 소수점(floating point), 고정 소수점(fixed point) 의 용어의 의미를 정확히 알아두자.



# 문자열. p53

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



- 큰 따옴표("...") 또는 작은 따옴표('...')로 텍스트를 감싸면 문자열이 된다.

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

```
>>> greeting
```

```
'Merry Christmas!'
```

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

```
>>> greeting
```

```
'Happy New Year!'
```





## 문자열. p54

- “...” 형태의 문자열 안에 “...” 형태의 문자열이 포함되면 컴파일러가 혼동한다. 이때는 “...” 형태의 문자열 안에 ‘...’ 형태의 문자열을 넣어 주면 된다. 반대로 하여도 된다.

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

- 여러 줄로 이루어진 문자열도 입력이 가능하다. “““ ... ””” 또는 ““ ... ””” 를 사용한다.

```
>>> 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.
```



## 문자열. p54

- 문자열은 + 연산자로 합칠 수 있다. 이것을 문자열 접합(string concatenation)이라고 한다.

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

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



# 문자열 <-> 수치값 p55

- 타입이 다른 데이터를 +로 합치려고 하면 오류가 발생한다.
- str()을 이용하여 문자열로 변환한 후에 합쳐야 한다.

```
>>> "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'
```

- 문자열을 정수로, 부동소수점으로 변환하는 함수도 있다.

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



# 문자열의 반복. p55

- \* 연산자를 이용하여 반복시킬 수 있다.

```
>>> line = "=" * 50
```

```
>>> print(line)
```

```
=====
```

```
>>> message = "Congratulations! "
```

```
>>> print(message*3)
```

```
Congratulations! Congratulations! Congratulations!
```



# 문자열 메소드. p56

- `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. p56

- 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> |

- 이들 번호를 인덱스라고 하고, 인덱스를 이용하여 각 문자를 추출할 수 있다.

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



# 문자열 안에 변수 출력. p56

- f-문자열(f-string) : 문자열 안에 출력하고 싶은 변수를 중괄호로 감싸서 넣는 방법

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

```
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



# 문자열 안에 변수 출력. p56

- (추가) + 또는 , 를 쓰는 다른 방법들도 익혀두자. 자료형 변환 등이 필요한 것과 차이점도 알아두자.





# 변수의 세부 구현 사항, p57

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

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

- 변수에 저장되는 것은 실제 값이 아니고 객체를 참조하는 값(주소)이다.

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



# 변수의 세부 구현 사항. p58

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

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

# 변수 y에 변수 x의 참조값이 복사된다.

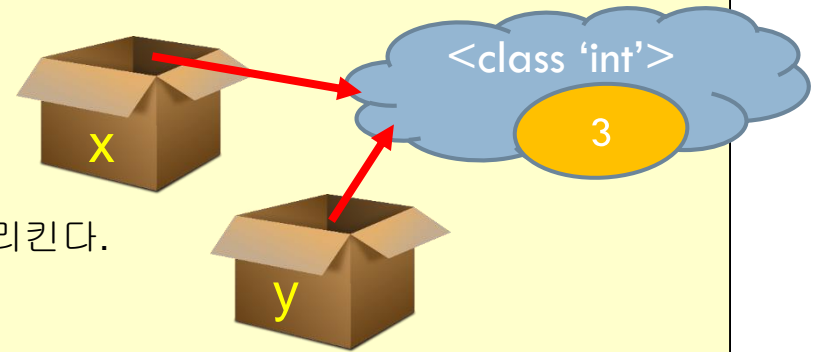
```
>>> id(x)
```

```
140721955779472
```

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

```
140721955779472
```

# 같은 주소를 가리킨다.

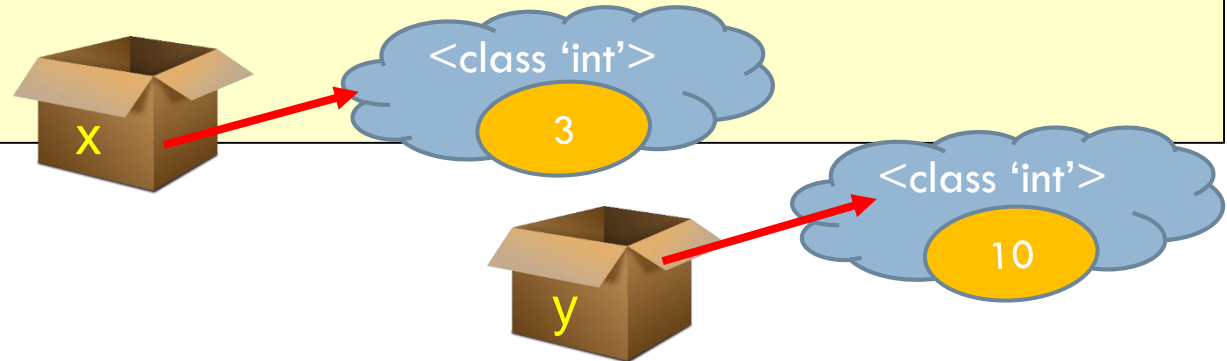


```
>>> y = 10
```

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

```
140721955779696
```

# 변수 y에 새로운 값이 할당되면 주소가 달라진다.



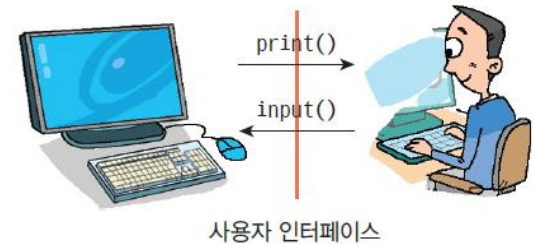


# input() 함수. p59

- input()
  - 주어진 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.



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

```
>>> name
홍길동
```

```
>>> print(name, "씨, 안녕하세요?")
홍길동 씨, 안녕하세요?
```



# 정수 입력. p60

```
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
```

```
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)
```

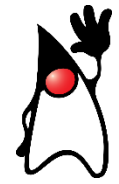
```
첫 번째 정수를 입력하시오: 300  
첫 번째 정수를 입력하시오: 400  
합은 700
```



# 중간점검. p61

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. p62

```
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. p63

- 주석(comment)

- 프로그래밍이 하는 일을 설명. 프로그램의 실행 결과에 영향을 끼치지 않는다. #로 시작하면 줄의 끝까지 주석으로 취급한다.

```
##  
#           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.
```

- 용도가 하나 더 -> 실행하고 싶지 않은 문장이 있다면 주석으로

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

합은 300



## 상수. p64

- 상수는 한번 값이 결정되면 절대로 변경되지 않는 변수
- 상수는 대문자로 하여서 변수와 구분하는 것이 보통이다.

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

- 왜 소스에 0,35라고 적지 않고 TAX-RATE처럼 적는 것일까?
  - 0.35보다는 TAX\_RATE으로 적으면 소스가 더 쉽게 읽힌다. 상수 이름이 주석의 역할을 한다.
  - 나중에 변경할 필요가 있다면 한 곳만 고치면 되므로 변경이 쉬워진다.

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





# Lab: 로봇 기자 만들기. p66

- 기사의 틀을 만들어 두고, 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 8:7로 이겼습니다.

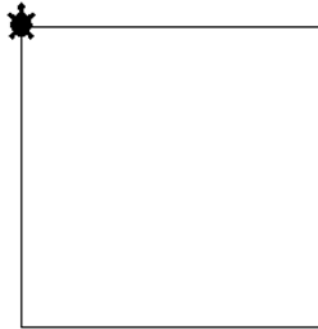
=====



# Lab: 사각형 그리기. p67

- 사용자로부터 사각형의 크기를 입력받아서 크기에 맞는 사각형을 그려보자.

사각형의 크기는 얼마로 할까요? 200





# Mini Project 거북이 조종하기. p69

- 사용자가 실시간으로 명령을 주어서 거북이를 조종하는 프로그램을 작성해보자. 이를 위하여 아직 학습하지 않았지만 다음과 같은 무한 반복 구문을 잠시 빌려서 사용하자.

```
while True:
```

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

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.

...



연습문제. p71



# Programming. p73

