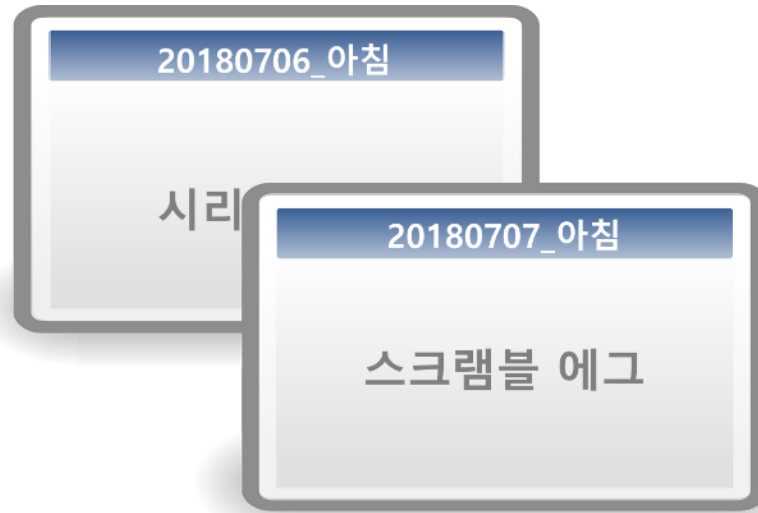


02

변수

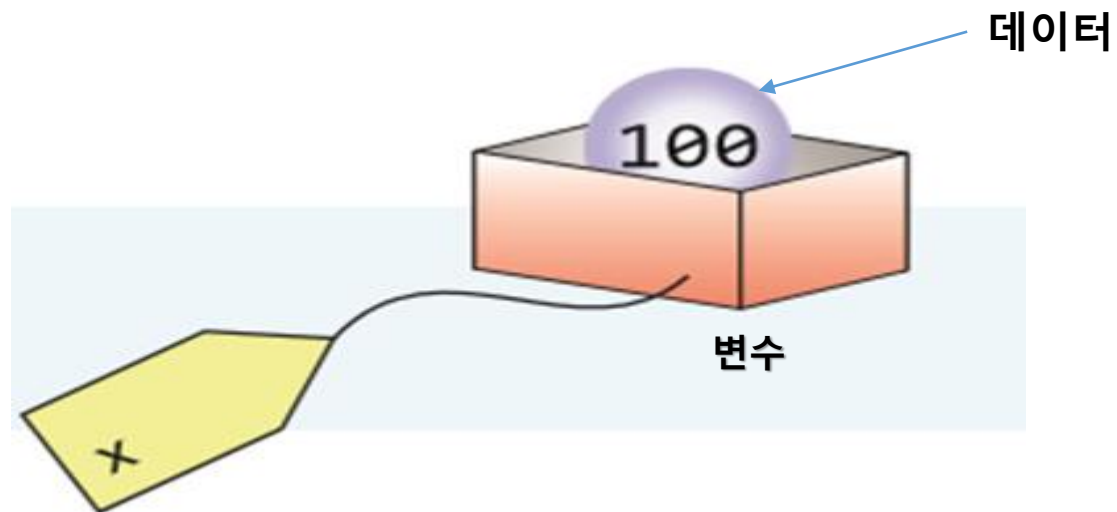


변수 = 값을 담는 바구니

01. 변수의 이해 - 변수 생성

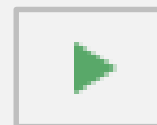
3 |

- 파이썬에서 변수를 생성하려면 다음과 같이 한다.



변수명 = 저장할 값

```
x = 100  
print(x)
```

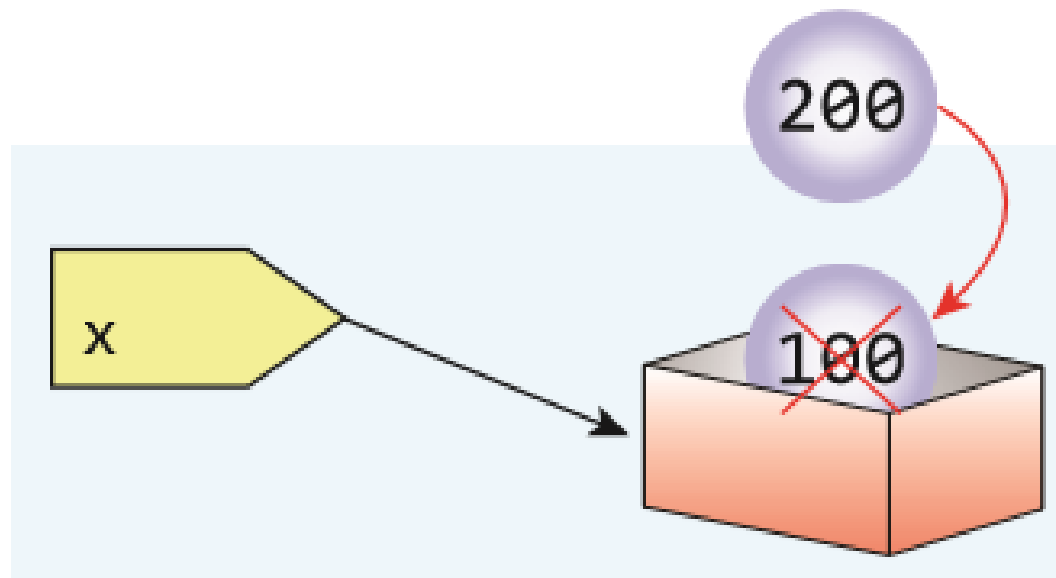


100

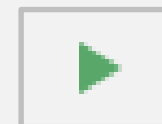
01. 변수의 이해 - 변수의 사용

4 |

- 생성된 변수에는 얼마든지 다른 값을 저장할 수 있다.



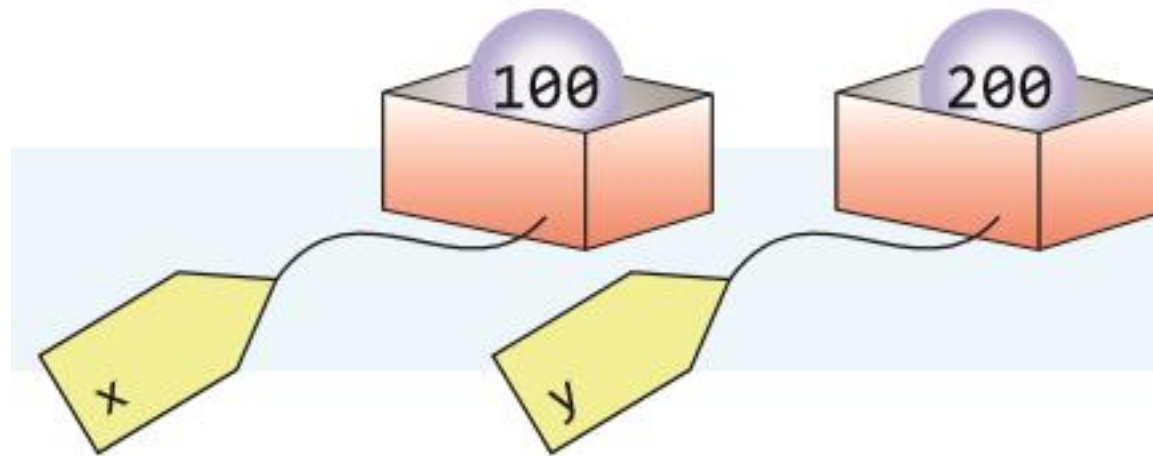
```
x = 100  
x = 200  
print(x)
```



200

01. 변수의 이해 - 변수 2개 생성

5 |



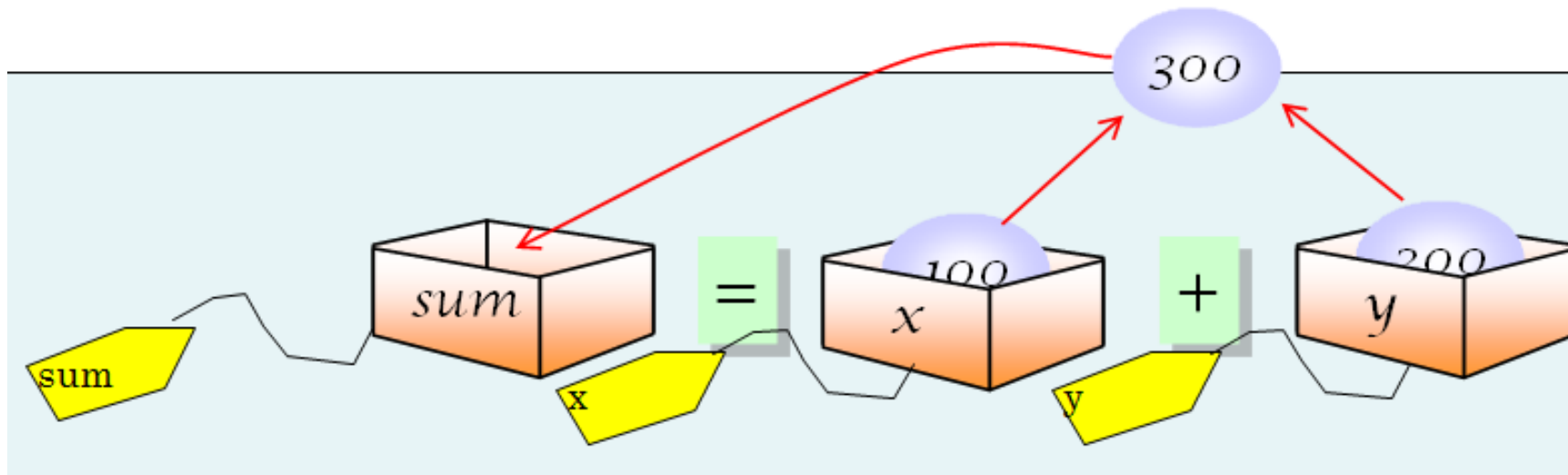
```
x = 100  
y = 200  
print(x, y)
```



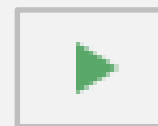
```
100 200
```

01. 변수의 이해 - 변수를 이용한 계산

6 |



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



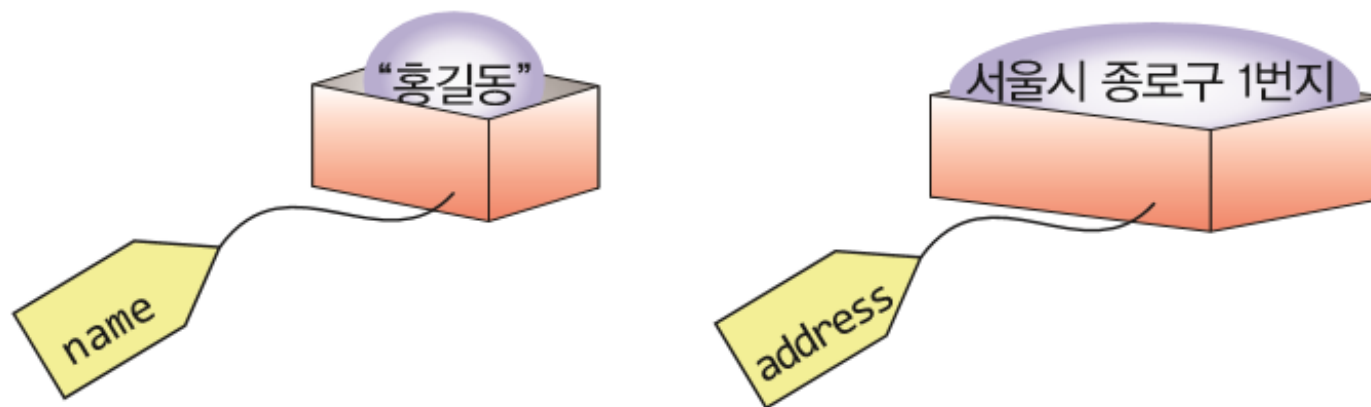
300

01. 변수의 이해 - 문자열도 변수에 저장할 수 있다!

71

- 파이썬의 변수에는 정수뿐만 아니라 문자열도 저장할 수 있다.

```
name = "홍길동"  
address = "부산시 부산진구 100"
```



```
print(name)  
print(address)
```



```
홍길동  
부산시 부산진구 100
```

- 'b'라는 변수를 설정하고 음수 3을 저장한 후 출력하여 보자.

```
b = -3  
print(b)
```


- 자신의 이름을 변수명으로 한 후 자신의 나이를 할당하여 출력하여 보자.

```
hong = 25  
print(hong)
```

■ 변수와 값

- 다음 코드를 살펴보자. `print(a + b)`와 `print("a + b")`의 차이는 따옴표의 사용 여부에 따른 의미 차이에 있다.

```
a=7  
b=5  
print(a+b)  
a=7  
b=5  
print("a+b")
```



```
12  
a+b
```

코드	의미
<code>print(a + b)</code>	a 변수에 있는 값과 b 변수에 있는 값을 더해 화면에 출력하라.
<code>print("a + b")</code>	"a + b"의 값을 그대로 화면에 출력하라.

[따옴표의 사용 여부에 따른 의미 차이]

■ 무엇이 출력될까?

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

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



■ 무엇이 출력될까?

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

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



13

76

■ 무엇이 출력될까?

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

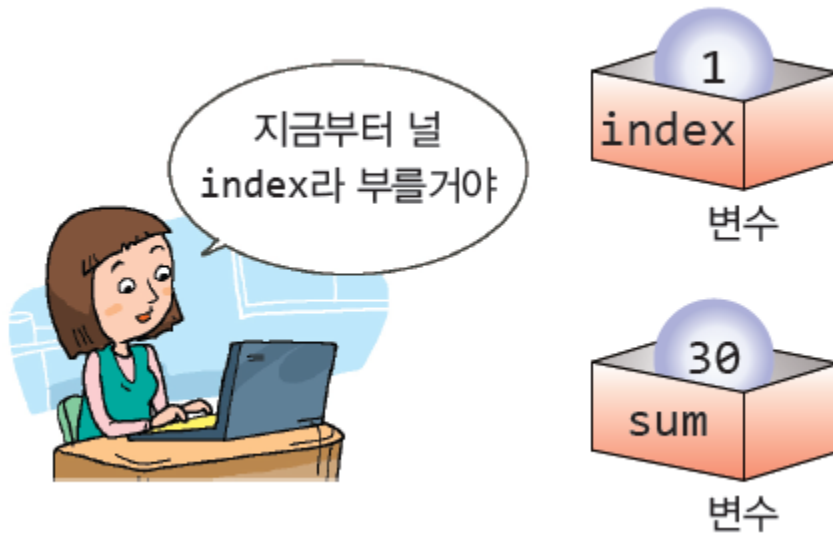
```
x = '7'  
y = '6'  
print(x+y)
```



13

76

- 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
- 특별한 의미가 있는 예약어는 사용할 수 없다.



```
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.
```

01. 변수의 이해

■ 변수이름 규칙

가능한 것 찾아보기

aBC	ab!	aB1	1ab
_ab	AB_C	1_	에이비

01. 변수의 이해

■ 변수이름 규칙

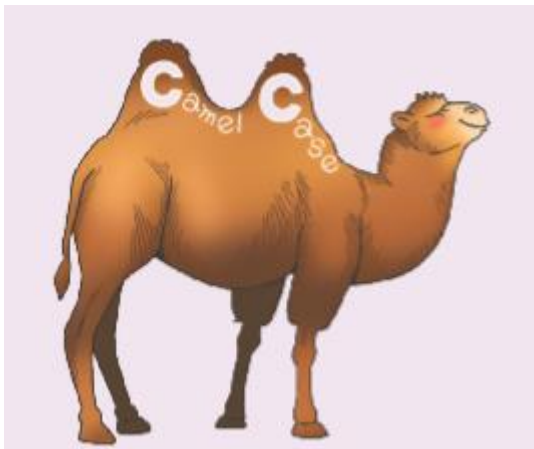
가능한 것 찾아보기

aBC	ab!	aB1	1ab
_ab	AB_C	1_	에이비

```
import keyword  
print(keyword.kwlist)
```

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

- 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



My new car

→ My_new_car → my_new_car

→ Mynewcar → myNewCar

```
print(type(변수명))
```

```
a=176  
print(type(a))  
  
print(type(176.9))
```

```
<class 'int'>  
<class 'float'>
```

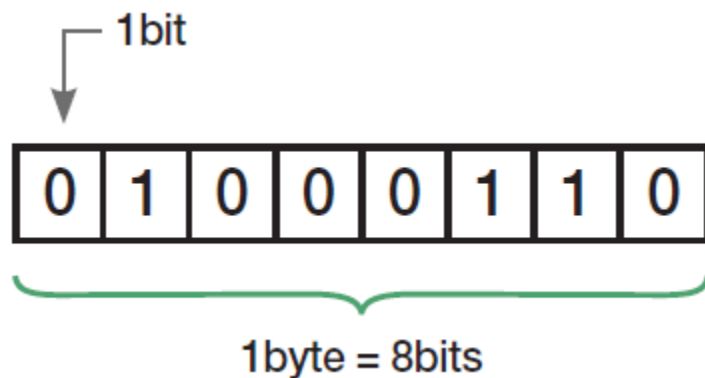
```
print(type(변수명))
```

```
b="키는 height"  
print(type(b))
```

```
<class 'str'>
```

■ 메모리 공간

- 하나의 변수를 메모리에 저장할 때, 그 변수의 크기만큼 공간(일정한 용량)을 할당받는다.
- 이진수 한 자리를 비트(bit)라고 하며, 8개의 비트는 1바이트(byte), 1,024바이트는 1킬로바이트(kilobyte, KB), 1,024킬로바이트는 1메가바이트(megabyte, MB)이다. 이러한 개념을 메모리 공간이라고 한다.



[비트(bit)와 바이트(byte)]

■ 기본 자료형

- 정수형(integer type) : 자연수를 포함해 값의 영역이 정수로 한정된 값.
- 실수형(floating-point type) : 소수점이 포함된 값.
- 문자형(string type) : 값이 문자로 출력되는 자료형.
- 불린형(boolean type) : 논리형으로, 참(True) 또는 거짓(False)을 표현할 때 사용

유형	자료형	설명	예	선언 형태
수치형	정수형	양수와 정수	1, 2, 3, 100, -9	data = 1
	실수형	소수점이 포함된 실수	10.2, -9.3, 9.0	data = 9.0
문자형	문자형	따옴표에 들어가 있는 문자형	abc, a20abc	data = 'abc'
논리형	불린형	참 또는 거짓	True, False	data = True

■ 기본 자료형

```
a = 1          #정수형  
b = 1          #정수형  
print(a,b)
```

1 1

```
a = 1.5         #실수형  
b = 3.5         #실수형  
print(a,b)
```

1.5 3.5

```
a = "ABC"       #문자형  
b = "101010"    #문자형  
print(a,b)
```

ABC 101010

```
a = True        #불린형  
b = False       #불린형  
print(a,b)
```

True False

■ 간단한 연산 : 사칙연산

- 덧셈 기호(+), 뺄셈 기호(-), 곱셈 기호(*), 나눗셈 기호(/)

```
print(25+30)
```

```
print(30-12)
```

```
print(50*3)
```

```
print(30/5)
```

```
55
```

```
18
```

```
150
```

```
6.0
```

■ 간단한 연산 : 제곱승

- 2개의 별표 기호(**)

```
print(3*3*3*3*3)    #3을 다섯번 곱합
```

```
print(3**5)         #3의 5승
```

```
243
```

```
243
```

```
number_1 = 2 ** 4  
number_2 = 4 ** 6  
  
print("number_1:", number_1)  
print("number_2:", number_2)
```

```
number_1: 16  
number_2: 4096
```

```
number_1 = 3.1 ** 4.2  
number_2 = 2.6 ** 5.4  
  
print("number_1:", number_1)  
print("number_2:", number_2)
```

```
number_1: 115.80281433592612  
number_2: 174.12297512053922
```

```
number_1 = 26
number_2 = 2 * 5 ** 2

print("number_1:", number_1)
print("number_2:", number_2)
```

```
number_1: 26
number_2: 50
```

■ 간단한 연산 : 나눗셈의 몫과 나머지 산출 연산

- 몫을 반환하는 연산자는 2개의 빗금 기호(`//`), 나머지 연산자는 백분율 기호(`%`)

```
number_1 = 7 // 3    #7 나누기 3의 몫  
number_2 = 7 % 3     #7 나누기 3의 나머지
```

```
print("number_1:", number_1)  
print("number_2:", number_2)
```

```
number_1: 2  
number_2: 1
```

■ 간단한 연산 : 증가 연산과 감소 연산

- 증가 연산자는 **+=**이고, 감소 연산자는 **-=**이다.

```
a = 1      # 변수 a에 1을 할당
```

```
a = a+1    # a에 1을 더한 후 그값을 a에 할당  
print(a)
```

```
a += 1     # a 증가연산  
print(a)
```

```
a = a+2    # a에 2를 더한 후 그값을 a에 할당  
print(a)
```

```
a += 2     # a 2 증가연산  
print(a)
```

2
3
5
7

```
a = 10     # 변수 a에 10을 할당
```

```
a = a-1    # a에 1을 뺀 후 그값을 a에 할당  
print(a)
```

```
a -= 1     # a 감소연산  
print(a)
```

```
a = a-2    # a에 2를 뺀 후 그값을 a에 할당  
print(a)
```

```
a -= 2     # a 2 감소 연산  
print(a)
```

9
8
6
4

input() 사용법

변수

사용자가 입력한 문자열을 숫자로 변환한다.

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

안내 메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 받는다.

```
x1 = input("첫 번째 숫자를 입력하세요: ")  
x2 = input("두 번째 숫자를 입력하세요: ")  
print(x1+x2)
```



문자열로 입력받음

```
x1 = int(input("첫 번째 숫자를 입력하세요: "))  
x2 = int(input("두 번째 숫자를 입력하세요: "))  
print(x1+x2)
```



정수(int)로 입력받음

문제되는 상황

프로그램에서 변수에 직접 숫자를 지정하는 경우는 정수(int)형으로 인식되지만, 사용자들로부터 숫자를 입력 받는 경우는 **스트링(string)형**으로 인식됨

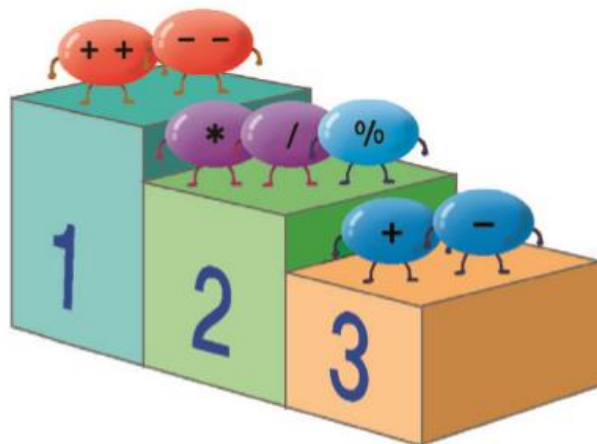
프로그램에서 자료형이 분명하게 표현되는 경우는 매우 드뭄
모든 변수에 대해서는 type함수를 이용하여 자료형을 확인하는 절차를 거쳐야 함

02. 자료형과 기본 연산

■ 연산자의 우선순위

$$x + \underbrace{y * z}_{\textcircled{1}}_{\textcircled{2}}$$

$$\underbrace{(x + y) * z}_{\textcircled{1}}_{\textcircled{2}}$$



02. 자료형과 기본 연산

■ 연산자의 우선순위표

순위	연산자	설명
1	**	지수 연산자
2	~ + -	단항 연산자
3	* / % //	곱셈, 나눗셈, 나머지 연산자
4	+ -	덧셈, 뺄셈
5	>> <<	비트 이동 연산자
6	&	비트 AND 연산자
7	^	비트 XOR 연산자, 비트 OR 연산자

■ 주석

- 주석(comment)은 소스 코드에 붙이는 설명글과 같은 것이다.

한줄 : *# 을 사용*

여러줄 : *''' ~ ''' 을 사용*

PyCharm 주석제거

- **ctrl + /** -> 주석처리, 주석 해제 둘 다 가능
- Mac 이라면 **command + /**

■ 주석

```
'''  
name = "홍길동"  
address = "서울시 종로구 1번지"  
print(name)  
print(address)  
'''  
  
# name = "홍길동"  
# address = "서울시 종로구 1번지"  
# print(name)  
# print(address)
```

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
x = 1
print("x:", x)
x += 1
print("x:", x)
x -= 1
print("x:", x)
```

```
x: 1
x: 2
x: 1
```

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
number = 5  
print("number:", number)
```

```
number = 4 * number + 6  
print("number:", number)
```

```
number: 5  
number: 26
```

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
count = 0
print("count:", count)
count += 1
print("count:", count)
count *= 2
print("count:", count)
```

```
count: 0
count: 1
count: 2
```

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
number = 2
print("number:", number)
number *= 5
print("number:", number)
number -= 4
print("number:", number)
number //= 2
print("number:", number)
number %= 2
print("number:", number)
```

```
number: 2
number: 10
number: 6
number: 3
number: 1
```


- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
a = 2 * 3 ** (1+2)
print(a)
a = 3 ** 4 / (2 + 2)
print(a)
a = 7 // 4 + 1 ** 5
print(a)
a = 6 + 11 * 2 // 4 ** 1
print(a)
a = ((3 + 4) ** 2) % 5 * 4
print(a)
```

```
54
20.25
2
11
16
```

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
number1 = 5  
number2 = number1  
number1 += number2  
print(number1)
```

```
10
```

PRACTICE

- 자료형과 기본연산

a,b에 각각 5,7 을 할당하고 곱셈을 한 값을 변수 result에 저장하여 출력하는 프로그램을 작성하라.

ANSWER

• 자료형과 기본연산

a,b에 각각 5,7 을 할당하고 곱셈을 한 값을 변수 result에 저장하여 출력하는 프로그램을 작성하라.

```
a = 5
b = 7
result = a*b
print('a*b:',result)
```

• 자료형과 기본연산

a,b에 각각 11,22를 할당하고, 그 후 숫자를 직접 이용하지 않고 다른 변수 c를 이용하여 두 수를 교환하는 프로그램을 작성하라.

ANSWER

• 자료형과 기본연산

a,b에 각각 11,22를 할당하고, 그 후 숫자를 직접 이용하지 않고 다른 변수 c를 이용하여 두 수를 교환하는 프로그램을 작성하라.

```
a = 11
b = 22
print('a:',a,'b:',b)
c = a
a = b
b = c
print('a:',a,'b:',b)
```

PRACTICE

• 자료형과 기본연산

다음 식의 몫과 나머지를 각각 다른 변수 `portion(몫)`, `the_rest(나머지)`에 저장 시키는 프로그램을 작성하라.

$123456 \div 789$

ANSWER

• 자료형과 기본연산

다음 식의 몫과 나머지를 각각 다른 변수 `portion`(몫), `the_rest`(나머지)에 저장 시키는 프로그램을 작성하라.

$123456 \div 789$

```
a = 123456
b = 789
portion = a//b
the_rest = a%b
print('portion:',portion,'the_rest:',the_rest)
```


- 자료형과 기본연산

반지름(r)이 7인 원의 넓이(area)를 제곱을 이용하여 구하는 프로그램을 작성하라 ($\pi=3.14$)

ANSWER

• 자료형과 기본연산

반지름(r)이 7인 원의 넓이(area)를 제곱을 이용하여 구하는 프로그램을 작성하라 ($\pi=3.14$)

```
r = 7
pi = 3.14
area = pi * r ** 2
print('반지름(r):', r, '넓이(area):', area)
```

• 자료형과 기본연산

number1, number2, number3, number4에 각각 5,4,9,2를 할당하여 아래 의 식을 계산하는 프로그램을 작성하라.

$$(number1 * number2) * (number2 - number3) * (number3 // number4)$$

ANSWER

• 자료형과 기본연산

number1, number2, number3, number4에 각각 5,4,9,2를 할당하여 아래 의 식을 계산하는 프로그램을 작성하라.

$$(number1 * number2) * (number2 - number3) * (number3 // number4)$$

```
number1 = 5
number2 = 4
number3 = 9
number4 = 2
```

```
result = (number1*number2)*(number2-number3)*(number3//number4)
print('result:',result)
```

03

문자형 자료형

■ 문자들의 집합 문자들의 연속

"Boy" "I am a boy"

```
tom = 'Boy'  
print(tom)  
bob = 'I am a boy'  
print(bob)
```

```
Boy  
I am a boy
```

- String형은 따옴표인 Quote를 이용하여 나타냄
- Python에서는 string형을 만들 때 다음 4가지 형식의 Quote 가 모두 허용됨

' '	" "	' ' ' '	'' ''
-----	-----	---------	-------

' '	string 내에 " " 를 포함해야 하는 경우
" "	string 내에 ' ' 를 포함해야 하는 경우
' ' ' '	여러 문장을 사용하고, string 내에 " " 를 포함해야 하는 경우
'' ''	여러 문장을 사용하고, string 내에 ' ' 를 포함해야 하는 경우

- quote의 형태에 따라 쓰임새를 주의해야 함

"He is a smart boy." my Teacher said

```
print('"He is a smart boy." my Teacher said')
```

"He is a smart boy." my Teacher said

- quote의 형태에 따라 쓰임새를 주의해야 함

He's a smart and diligent boy.

```
print("He's a smart and diligent boy.")
```

```
He's a smart and diligent boy.
```

- quote의 형태에 따라 쓰임새를 주의해야 함

He's a smart and diligent boy.

```
print('He's a smart and diligent boy.')
```

```
print('He's a smart and diligent boy.')
```

```
File "D://_Works/PyCharm/Test001/Test.py", line 1
```

```
    print('He's a smart and diligent boy.')
```

```
        ^
```

```
SyntaxError: invalid syntax
```

```
Process finished with exit code 1
```

- '+' 연산자를 두 문자(열) 사이에 사용하여 둘을 합침

"문자(열)" + "문자(열)" + ... + "문자(열)"

각 문자 또는 문자열을 병합하여 하나의 문자로 생성

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
name = 'Gildong'  
location = 'Busan'  
print('Hello, My name is ' + name + '. And I live in ' + location + '.')
```

```
Hello, My name is Gildong. And I live in Busan.
```

- '*' 연산자를 사용해서 같은 문자(열)를 반복적으로 출력할 수 있음

"문자(열)" * n (n은 정수)

문자 또는 문자열을 n의 배수만큼 반복하여 하나의 문자로 생성

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
start = '=' * 10  
title = 'Python Program'  
finish = '=' * 10  
print(start + title + finish)
```

```
=====Python Program=====
```

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
string1 = 'Hello,'  
string2 = ' Python'  
star = '***'  
print(star * 3 + string1 + string2 + star*3)
```

```
*****Hello, Python*****
```

03. String 형 - 문자열 길이를 알려주는 함수:: len()

- Python에서는 공백(space)도 문자로 취급한다.
- 길이 측정 시 공백(space) 또한 문자(열)의 길이(크기)에 속함

len(문자 또는 문자열)

문자 또는 문자열의 길이(크기)를 반환

I		a	m		a		b	o	y
---	--	---	---	--	---	--	---	---	---

10 글자

03. String 형 – 문자열 길이를 알려주는 함수:: len()

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
string1 = 'Hello World'
string2 = 'Good Morning'
print("Len of",string1,":",len(string1))
print("Len of",string2,":",len(string2))
print(len(string1)*len(string2))
```

```
Len of Hello World : 11
Len of Good Morning : 12
132
```

03. String 형 – 문자열 길이를 알려주는 함수:: len()

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
string1 = 'python'  
string2 = 'programming'  
print(len(string1), len(string2))  
string3 = (string1 + ' ' + string2)  
print(len(string3))
```

```
6 11  
18
```

03. String 형 – String의 indexing

- string의 index는 string내의 문자들에 대해 순번을 부여하는 것과 같음
- 공백(space)또한 index가 부여된다.

string의 변수명[index_num]	index_num에 해당하는 문자를 반환 (index는 항상 0부터 시작) 단, index_num으로 음수를 입력하면 string의 끝에서부터 문자를 반환
------------------------	---

0	1	2	3	4	5	6	7
m	o	u	n	t	a	i	n

string의 indexing을 활용하여 주어진 문장으로부터 새로운 단어를 합성해 보자.

- 1) 변수 tom은 'Tom is a good student.' string을 담고 있다.
- 2) 새로 합성할 단어는 'mood'이다.

```
tom = 'Tom is a good student.'  
firstIndex = tom[2]  
secondIndex = tom[10]  
thirdIndex = tom[11]  
fourthIndex = tom[12]  
print(firstIndex + secondIndex + thirdIndex + fourthIndex)
```

03. String 형 – String의 indexing

- 다음 명령문을 직접 실행시켜본 후 결과를 확인해 보자

```
string1 = 'Beautiful Weather'  
firstindex = string1[0]  
secondindex = string1[1]  
thirdindex = string1[11]  
print(firstindex + secondindex + thirdindex)
```

Bee

03. String 형 – immutable(변경불가)

■ string은 한 번 지정하면 더 이상 바꿀 수 없음

1. 할당문(=)을 지원하지 않음
2. index를 이용한 변경을 허용하지 않는다. 즉, index를 이용한 할당은 에러가 된다.

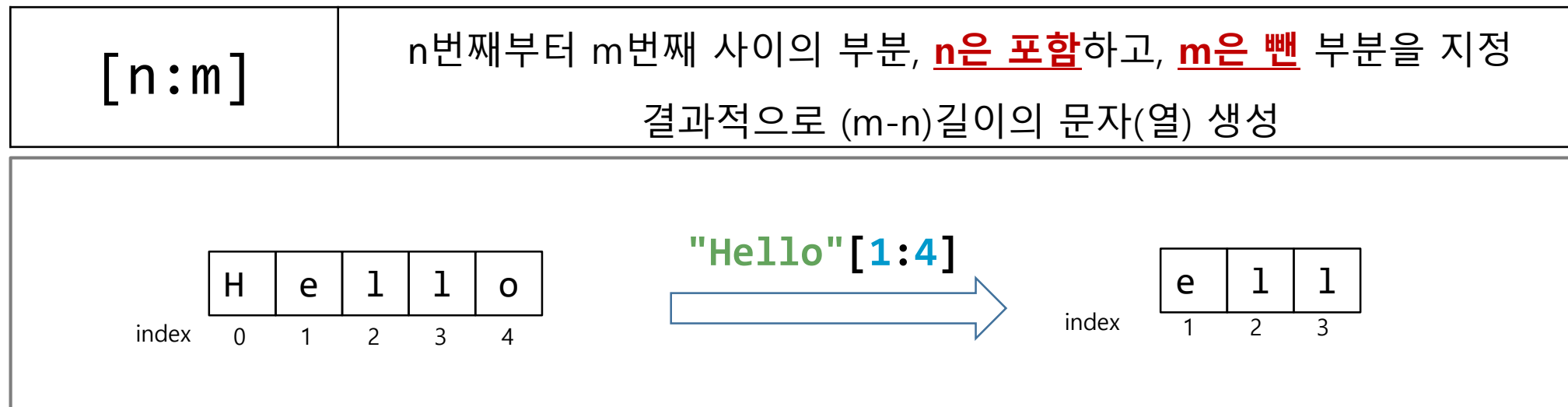
```
greeting = "Hello, world!" 일 때  
greeting[0] = 'J'를 실행 시킨 후 결과를 확인해 보자.
```

```
greeting = "Hello, world!"  
greeting[0] = 'J'  
print(greeting)
```

```
Traceback (most recent call last):  
  File "D:/_Works/PyCharm/Test001/Test.py", line 2, in <module>  
    greeting[0] = 'J'  
TypeError: 'str' object does not support item assignment
```

03. String 형 – slice

- slice를 활용하면 string의 일부를 취할 수 있음



string의 slice를 활용하여 새로운 단어를 생성해보자.

- 1) 변수 tom은 "Tom is a good student." 문자열을 담고 있다.
- 2) 일부를 추출할 단어는 "good" 이다.

```
tom = 'Tom is a good student.'  
print(tom[9:13])
```

good

- slice의 범위를 지정하는 n과 m은 string의 index범위 내의 양의 정수로 정해져야 한다.
- 지정하지 않은 slice 범위에 대해서는 기본값이 적용됨
 1. string의 처음과 끝에 해당하는 index로 자동 지정

[:n]	처음부터 n을 포함하지 않는 범위까지 지정해서 slice
[m:]	m번째부터 string의 끝까지 범위를 지정해서 slice
[m:문자열보다 큰 값]	m번째부터 string의 끝까지 범위를 지정해서 slice

slice를 활용하여 주어진 string을 여러가지 형태로 분리 해 보자.

1) 변수 fruit은 "banana" 문자열을 담고 있다.

```
fruit = 'banana'
print(fruit[:3])
'ban'
print(fruit[3:])
'ana'
print(fruit[3:999])
'ana'
```


• String - slice

Indexing과 slice를 활용하여 다음 조건을 만족하는 문자열을 출력해 보자

- 1) 변수 string1은 "red apple" 문자열을 담고 있다.
- 2) 변수 string2는 "yellow banana" 문자열을 담고 있다.
- 3) "yellow apple" 과 "red banana" 문자열을 생성해보자

ANSWER

• String - slice

Indexing과 slice를 활용하여 다음 조건을 만족하는 문자열을 출력해 보자

- 1) 변수 string1은 "red apple" 문자열을 담고 있다.
- 2) 변수 string2는 "yellow banana" 문자열을 담고 있다.
- 3) "yellow apple" 과 "red banana" 문자열을 생성해보자

```
string1 = "red apple"  
string2 = "yellow banana"  
print(string2[:6]+string1[3:], string1[:3]+string2[6:])
```

- **String - slice**

다음 이름 "Hong Gildong" 가운데 family name인 "Hong" 을 출력하는 프로그램을 작성하라

- String - slice

다음 이름 "Hong Gildong" 가운데 family name인 "Hong" 을 출력하는 프로그램을 작성하라

```
name = "Hong Gildong"  
family_name = name[:4]  
print(family_name)
```

04

자료형 변환

■ 정수형과 실수형 간 변환

- `float()` 함수 : 정수를 실수형으로 변환해 주는 함수.

```
a = 10
print(a)
```

#a 변수에 정수 데이터 10을 할당
#a가 정수형으로 출력

```
a = float(a)
print(a)
```

#a를 실수형으로 변환
#a를 실수형으로 출력

10

10.0

■ 정수형과 실수형 간 변환

- `float()` 함수 : 정수를 실수형으로 변환해 주는 함수.

```
a = 10          #a에 정수 데이터 10 할당
b = 3           #b에 정수 데이터 3 할당

print(a/b)      #실수형으로 a나누기 b를 출력
```

```
3.3333333333333335
```

■ 정수형과 실수형 간 변환

- `int()` 함수 : 실수를 정수형으로 변환해 주는 함수.

```
a = 10.0  
print(a)
```

#a 변수에 실수 데이터 10.0을 할당
#a가 실수형으로 출력

```
a = int(a)  
print(a)
```

#a를 정수형으로 변환
#a를 정수형으로 출력

```
10.0  
10
```


■ 정수형과 실수형 간 변환

- `int()` 함수 : 실수를 정수형으로 변환해 주는 함수.

```
a = int(10.7)
b = int(10.3)
print(a+b)      # 정수형 a와 b의 합을 출력
print(a)        # 정수형 a값 출력
print(b)        # 정수형 b값 출력
```

```
20
10
10
```

■ 숫자형과 문자형 간 변환

- 실수형 값을 문자형으로 선언하기 위해서는 반드시 따옴표를 붙여 선언해야 한다.

```
a = '76.3'          #a에 문자열 76.3을 할당, 문자열을 의미
b = float(a)        #a를 실수형으로 변환 후 b에 할당
print(a)            #a값 출력
print(b)            #b값 출력
```

```
76.3
76.3
```

```
a = '76.3'          #a에 문자열 76.3을 할당, 문자열을 의미
b = float(a)        #a를 실수형으로 변환 후 b에 할당
print(a + b)
```

```
File "D:/_Works/PyCharm/Test001/Test.py", line 3, in <module>
    print(a + b)
TypeError: can only concatenate str (not "float") to str
```

■ 숫자형과 문자형 간 변환

- 두 변수를 더하기 위해서는 다음과 같이 두 변수의 자료형을 통일해야 한다.

```
a = '76.3'
a = float(a)      # a를 실수형으로 변환 후 a에 할당
b = a             # 실수형 a값을 b에 할당
print(a+b)        # 두 실수형을 더한 후 출력
```

152.6

■ 숫자형과 문자형 간 변환

- **str() 함수** : 기존의 정수형이나 실수형을 문자열로 바꿔 준다.
문자형 간의 덧셈은 숫자 연산이 아닌 단순 붙이기(concatenate)가 일어난다.

```
a = b = 76.3
a = str(a)           # 실수형 a값을 문자열로 변환 후 a할당
b = str(b)           # 실수형 b값을 문자열로 변환 후 b할당
print(a+b)           # 두 값을 더한 후 출력
```



```
76.376.3             # 문자열 간 덧셈은 문자열 간 단순연결
```

■ 자료형 확인하기

- `type()` 함수 : 자료형을 확인할 수 있는 함수.

```
a = int(10.3)          # a는 정수형으로 10.3을 할당
b = float(10.3)        # b는 실수형으로 10.3을 할당
c = str(10.3)          # c는 문자형으로 10.3을 할당

print(type(a))         # a의 타입을 출력
print(type(b))         # b의 타입을 출력
print(type(c))         # c의 타입을 출력
```

```
<class 'int'>
<class 'float'>
<class 'str'>
```

CH 03

화면 입출력

01. 화면 입출력

■ 표준 입력 함수: input() 함수

- input() 함수: 표준 입력 함수로, 사용자가 문자열을 콘솔 창에 입력할 수 있게 해 준다.

```
print("Enter your name:")  
somebody = input()  
print("Hi",somebody, "How are you today?")
```

```
Enter your name:  
Gildong  
Hi Gildong How are you today?
```

01. 화면 입출력

■ 표준 입력 함수: input() 함수

<code>variable_name=input()</code>	사용자로부터 입력을 받는다.
<code>variable_name=input('문자열')</code>	'문자열'에 해당하는 내용을 출력 후 사용자로부터 입력을 받는다.

01. 화면 입출력

■ 표준 출력 함수: print() 함수

- **print() 함수:** 표준 출력 함수로, 결과를 화면에 출력하는 함수이다.

```
print("Hello World!", "Hello Again!!!")
```

```
Hello World! Hello Again!!!
```

PRACTICE

- 변수 `height` 에 키(tall)를 묻고 값을 할당 하시오

키를 입력하시오: 180

당신의 키는 180.0cm 입니다.

ANSWER

- 변수 `height` 에 키(tall)를 묻고 값을 할당 하시오

키를 입력하시오: 180

당신의 키는 180.0cm 입니다.

```
height = float(input("키를 입력하시오: "))
print("당신의 키는 "+str(height)+"cm 입니다 ")
print("당신의 키는 %.1fcm입니다" %(height))
print("당신의 키는 {0:.1f}cm입니다".format(height))
```

```
height = float(input("키를 입력하시오: "))
print("당신의 키는 %fcm입니다" %(height))
print("당신의 키는 {}cm입니다".format(height))
```

PRACTICE

- 변수 `height` 에 키(`tall`)를 묻고 값을 할당 하시오

(키를 정수형으로 만들어 소수점 아래를 버리시오)

키를 입력하시오: `180.5`

당신의 키는 180cm 입니다.

ANSWER

- 변수 height 에 키(tall)를 묻고 값을 할당 하시오

(키를 정수형으로 만들어 소수점 아래를 버리시오)

키를 입력하시오: 180.5

당신의 키는 180cm 입니다.

```
height = int(float(input("키를 입력하시오: ")))  
print("당신의 키는 ", height, "cm 입니다 " )  
print("당신의 키는 %dcm입니다" %(height))  
print("당신의 키는 {}cm입니다".format(height))
```

01. 화면 입출력

■ 표준 출력 함수: print() 함수

- **print() 함수:** 표준 출력 함수로, 결과를 화면에 출력하는 함수이다.

```
temperature = float(input("온도를 입력하세요: "))  
print(temperature)
```

```
온도를 입력하세요: 30  
30.0
```

01. 화면 입출력

■ 표준 출력 함수: print() 함수

<code>print('문자열')</code>	'문자열'을 화면에 출력해준다.
<code>print(variable_name)</code>	변수 variable_name에 해당하는 값을 화면에 출력해 준다.
<code>print('문자열', variable_name)</code>	'문자열'과 변수 variable_name에 해당하는 값을 연속해서 화면에 출력해 준다.

01. 문자열의 이해

여기서 잠깐! 문자열 표현과 특수문자

- 파이썬에서 문자열을 표현할 때 작은따옴표나 큰따옴표를 사용한다. 하지만 다음과 같이 아포스트로피(')가 문장에 들어가면 작은따옴표를 사용하기 어렵다. 만약, 작은따옴표로 문자열을 표현한다면 인터프리터는 이 문자가 제대로 닫히지 않았다고 판단하고 오류를 출력할 것이다

```
It's OK.
```

- 이러한 문제를 지원하기 위해 파이썬에서는 다양한 기능을 제공한다. 먼저 문자열 자체에 작은따옴표나 큰따옴표가 들어가 있는 경우이다.

```
a = "It's OK."
```


01. 문자열의 이해

여기서  잠깐! 문자열 표현과 특수문자

- 다음으로 파이썬의 특수문자 기능을 사용하는 것이다. 아래의 특수문자를 사용할 경우 다음과 같이 아포스트로피(')를 사용할 수 있다.

특수문자	기능	특수문자	기능
<code>\</code> <code>Enter</code>	다음 줄과 연속임을 표현	<code>\b</code>	백스페이스
<code>\</code>	<code>\</code> 문자 자체	<code>\n</code>	줄 바꾸기
<code>\'</code>	' 문자	<code>\t</code>	<code>Tab</code> 키
<code>\"</code>	" 문자	<code>\e</code>	<code>Esc</code> 키

```
a = "It\'s OK."
```

01. 문자열의 이해

- String형은 따옴표인 Quote를 이용하여 나타냄
- Python에서는 string형을 만들 때 다음 4가지 형식의 Quote 가 모두 허용됨

' '	" "	' ' ' '	'''' ''''
-----	-----	---------	-----------

' '	string 내에 " " 를 포함해야 하는 경우
" "	string 내에 ' ' 를 포함해야 하는 경우
' ' ' '	여러 문장을 사용하고, string 내에 " " 를 포함해야 하는 경우
'''' ''''	여러 문장을 사용하고, string 내에 ' ' 를 포함해야 하는 경우

01. 문자열의 이해

여기서 잠깐! 문자열 표현과 특수문자

- 또 다른 문제로는 다음과 같은 줄 바꿈 표현이 있다. 이러한 경우에도 문자열로 표현하기 어렵다.
- 두 줄 이상의 표현도 마찬가지로 표현할 수 있다. 하나는 큰따옴표(")나 작은따옴표(')를 3개로 연결하는 방법이다. 다음과 같이 선언한다

```
a = """  
It's OK.  
I'm Happy.  
See you."""  
print(a)
```

```
It's OK.  
I'm Happy.  
See you.
```

03

문자열 서식 지정

03. 문자열 서식 지정

■ 서식 지정의 개념

- `print()` 함수를 사용하다 보면 어떤 형식에 맞추어 결과를 출력해야 할 일이 발생하기도 한다. 특히 엑셀을 사용할 때 통화 단위, 세 자리 숫자 단위 띄어쓰기, % 출력 등 다양한 형식에 맞추어 출력할 일이 생기는데, 이를 서식 지정(formatting)이라고 한다.

03. 문자열 서식 지정

■ % 서식과 format() 함수

- 문자열의 서식(format)을 설정할 때, print() 함수는 기본적인 출력 형식 외에 % 서식과 format() 함수를 구문으로 사용하여 출력 양식을 지정할 수 있다.

```
print(1,2,3)
print("a"+" "+"b"+" "+"c")
print("%d %d %d" %(1,2,3))
print("{}{}{}".format("a","b","c"))
```

```
1 2 3
abc
1 2 3
abc
```

03. 문자열 서식 지정

■ % 서식과 format() 함수

- ① 데이터와 출력 형식을 분류할 수 있다. 같은 내용을 여러 번 출력하기 위해 기존 `print()` 문에 띄어쓰기를 넣어 + 기호로 문자열 형태를 붙여 주는 것보다 시각적으로 훨씬 이해하기 쉽게 코드를 표현할 수 있다.
- ② 데이터를 형식에 따라 다르게 표현할 수 있다. [코드 6-3]을 보면 문자열 형태인 ('one', 'two') 구문과 정수형인 (1, 2) 구문이 각각 %s와 %d로 다르게 할당되는 것을 확인할 수 있다. 서식 지정 기능은 각 변수의 자료형에 맞게 다른 서식으로 지정한다

```
print('%s %s' %('one', 'two'))  
print('%d %d' %(1,2))
```

```
one two  
1 2
```

03. 문자열 서식 지정

■ % 서식과 format() 함수 : % 서식

- % 서식은 다음과 같은 형태로 출력 양식을 표현하는 기법이다.

'%자료형 % (값)'

```
print("I eat %d apples." %3)  
print("I eat %s apples." %"five")
```

```
I eat 3 apples.  
I eat five apples.
```


03. 문자열 서식 지정

■ % 서식과 format() 함수 : % 서식

서식	설명
%s	문자열(string)
%c	문자 1개(character)
%d	정수(integer)
%f	실수(floating-point)
%o	8진수
%x	16진수
%%	문자 % 자체

```
print("I love %s" %("meet"))
print("My thing is %dcm" %(180))
print("I get %c grade" %("A"))
print("%x마일 남았어" %(20))
print("%e키로만 더 찌워야지" %(12345678))
print("%f키로만 더 빼야지" %(0.005))
```

```
I love meet
My thing is 180cm
I get A grade
14마일 남았어
1.234568e+07키로만 더 찌워야지
0.005000키로만 더 빼야지
```

03. 문자열 서식 지정

■ % 서식과 format() 함수 : % 서식

- %는 1개 이상의 값도 할당할 수 있다. 다음 코드처럼 % 뒤에 괄호를 쓰고, 그 안에 순서대로 값을 입력하면 된다.

```
print("Product: %s, Price per unit: %f." %("Apple",5.243))
```

```
Product: Apple, Price per unit: 5.243.
```

03. 문자열 서식 지정

■ % 서식과 format() 함수 : % 서식

- 직접 값을 넣지 않고 number와 day 같은 변수명을 넣어도 문제없이 실행된다.

```
number = 3  
day = "three"  
print("I ate %d apples. I was sick for %s days." %(number, day))
```

```
I ate 3 apples. I was sick for three days.
```

■ 자주 사용하지 않는 이유

```
first_name = 'Jim'
last_name = 'Carry'
age = 34
profession = 'comedian'
affiliation = 'Monty Python'
text = 'Hello, %s %s. You are a %d. You are a %s. You were a member of %s.'
%(first_name, last_name, age, profession, affiliation)
print(text)
```

Hello, Jim Carry. You are a 34. You are a comedian. You were a member of Monty Python.

03. 문자열 서식 지정

■ % 서식과 format() 함수 : format() 함수

- **format() 함수** : % 서식과 거의 같지만, 문자열 형태가 있는 함수를 사용한다는 차이점이 있다. 문자열 서식은 함수이므로 다음과 같은 형태로 서식을 지정할 수 있다.

```
"{자료형}".format(인수)
```

- 다음 코드는 format() 함수를 사용한 가장 기본적인 표현 형태로, 숫자 20이 {0}에 할당되어 출력된다. 기존 % 서식과 비교하면, 자료형을 바로 지정해 주지 않고 순서대로 변수가 할당된다는 장점이 있다.

```
print("I'm {0} years old.".format(20))
```

```
I'm 20 years old.
```

03. 문자열 서식 지정

■ % 서식과 format() 함수 : format() 함수

- format() 함수는 % 서식처럼 변수의 이름을 사용하거나 변수의 자료형을 따로 지정하여 출력한다.

```
age = 23; name = 'Hong Gildong'
print("I'm {0} years old.".format(age))
print("My name is {0} and {1} years old.".format(name,age))
print("Product:{0},Price per unit:{1:.2f}.".format("Apple",5.243))
```

```
I'm 23 years old.
My name is Hong Gildong and 23 years old.
Product:Apple,Price per unit:5.24.
```

- ➡ 4행의 Price per unit: {1:.2f}는 기존 format() 함수의 쓰임과 다르게 .2f라는 구문이 추가되었다. 2는 소수점 둘째 자리까지 출력하라는 뜻이다.

```
test = 'Hello {}'.format('Bob')  
print(test)
```

Hello Bob

03. 문자열 서식 지정

■ 패딩

1. 파이썬의 서식 지정 기능에는 여유 공간을 지정하여 글자 배열을 맞추고 소수점 자릿수를 맞추는 패딩(padding)기능이 있다. % 서식과 format() 함수 모두 패딩 기능을 제공한다.

```
print("%10d"%12)  
print("%-10d"%12)
```

```
          12  
12
```

- ➡ 첫 번째 줄의 print("%10d" % 12)는 10자리의 공간을 확보하고, 우측 정렬로 12를 출력하라는 명령이다. 기본 정렬이 우측 정렬이므로 좌측에서 아홉 번째 칸부터 12가 출력된다. 좌측정렬을 하기 위해서는 세 번째 줄처럼 - 부호를 붙이면 된다.

03. 문자열 서식 지정

■ 패딩 : % 서식의 패딩

```
>>> print("%10.3f" % 5.94343)      # 10자리를 확보하고 소수점 셋째 자리까지 출력
      5.943
>>> print("%10.2f" % 5.94343)      # 10자리를 확보하고 소수점 둘째 자리까지 출력
      5.94
>>> print("%-10.2f" % 5.94343)
5.94
```

➡ 실수에서도 자릿수와 소수점 자릿수를 지정할 수 있다.

첫 번째 줄의 `print("%10.3f" % 5.94343)`은 10자리의 공간을 확보하고 소수점 셋째 자리까지 출력하라는 뜻이다. 이때 10자리 안에는 소수점이 포함된다. 역시 우측 정렬 기준이며, 좌측 정렬을 하기 위해서는 - 부호를 붙이면 된다.

03. 문자열 서식 지정

■ 패딩 : `format()` 함수의 패딩

```
>>> print("{0:>10s}".format("Apple"))
      Apple
>>> print("{0:<10s}".format("Apple"))
Apple
```

- ➡ 첫 번째 줄의 `print("{0:>10s}".format("Apple"))`은 10자리의 공간을 확보하고, 우측 정렬로 문자열 'Apple'을 출력하라는 명령이다. 좌측 정렬을 하기 위해서는 '`{0:<10s}`'처럼 < 부호를 사용하면 된다.

PRACTICE

- input 연습문제

당신의 나이는 몇 살입니까? 30
당신은 30 년을 살았습니다.

ANSWER

- input 연습문제

당신의 나이는 몇 살입니까? 30
당신은 30 년을 살았습니다.

```
a = input("당신의 나이는 몇 살입니까? ")  
print("당신은",a,"년을 살았습니다.")
```

PRACTICE

• input 연습문제

당신의 키는? 180

당신의 적정 몸무게는 72 kg 입니다.

적정 몸무게 : $(키 - 100) * 0.9$

ANSWER

• input 연습문제

당신의 키는? 180

당신의 적정 몸무게는 72 kg 입니다.

적정 몸무게 : $(\text{키}-100)*0.9$

```
a = int(input("당신의 키는? "))  
print("당신의 적정 몸무게는 ", int((a-100)*0.9), "kg 입니다.")
```

PRACTICE

• 나머지 연산자 연습문제

분자를 입력하시오: 7

분모를 입력하시오: 4

나눗셈의 몫 = 1 나머지 = 3 입니다.

ANSWER

• 나머지 연산자 연습문제

분자를 입력하시오: 7

분모를 입력하시오: 4

나눗셈의 몫 = 1 나머지 = 3 입니다.

```
a = int(input("분자를 입력하시오:"))  
b = int(input("분모를 입력하시오:"))  
print("나눗셈의 몫 =", a//b, "나머지 =", a%b, "입니다.")
```


PRACTICE

- 나머지 연산자 연습문제

초단위의 시간을 받아서 몇 분 몇 초인지를 계산하여 보자.

시간을 입력하시오(초단위) : 1000

1000 초는 0시간 16분 40초 입니다.

ANSWER

• 나머지 연산자 연습문제

초단위의 시간을 받아서 몇 분 몇 초인지를 계산하여 보자.

시간을 입력하시오(초단위) : 1000

1000 초는 0시간 16분 40초 입니다.

```
total_sec = int(input("시간을 입력하시오(초단위) :"))
hour = total_sec // 60**2
remain_sec = total_sec % (60**2)
min = remain_sec // 60
sec = remain_sec % 60
print(total_sec,"초는", hour,"시간",min,"분",sec,"초 입니다.")
print(hour*3600+min*60+sec)
```

PRACTICE

• 단순계산기(덧셈)

덧셈 첫 번째 숫자는? 5
덧셈 두 번째 숫자는? 8
두 숫자의 합은 13 입니다.

ANSWER

• 단순계산기(덧셈)

덧셈 첫 번째 숫자는? 5
덧셈 두 번째 숫자는? 8
두 숫자의 합은 13 입니다.

```
a = int(input("덧셈 첫 번째 숫자는? "))  
b = int(input("덧셈 두 번째 숫자는? "))  
print("두 숫자의 합은", a+b, "입니다.")
```

PRACTICE

• 평균계산기

첫 번째 숫자를 입력하세요: 10
두 번째 숫자를 입력하세요: 20
세 번째 숫자를 입력하세요: 30
세 수의 평균은 20 입니다.

ANSWER

• 평균계산기

첫 번째 숫자를 입력하세요: 10
두 번째 숫자를 입력하세요: 20
세 번째 숫자를 입력하세요: 30
세 수의 평균은 20 입니다.

```
a = int(input("첫 번째 숫자를 입력하세요:"))  
b = int(input("두 번째 숫자를 입력하세요:"))  
c = int(input("세 번째 숫자를 입력하세요:"))  
print("세 수의 평균은", (a+b+c)/3, "입니다.")
```

PRACTICE

• 단순계산기(나눗셈)

제수는? 13

피제수는? 5

나눗셈의 몫은 2 나머지는 3 입니다.

제수
—
피제수

ANSWER

• 단순계산기(나눗셈)

제수는? 13

피제수는? 5

나눗셈의 몫은 2 나머지는 3 입니다.

제수
—
피제수

```
a = int(input("제수는? "))  
b = int(input("피제수는? "))  
print("나눗셈의 몫은", a//b, "나머지는", a%b, "입니다.")
```


PRACTICE

• 커피가게 매출 계산하기

우리가 커피 전문점을 내려고 한다. 다음과 같이 커피 메뉴가 있을 때, 얼마나 매출을 올릴 수 있을지 계산해 보자

아메리카노 : 2,000원

카페라떼 : 3,000원

카푸치노 : 3,500원

아메리카노 판매 개수를 입력하세요 : 10

카페라떼 판매 개수를 입력하세요 : 20

카푸치노 판매 개수를 입력하세요 : 30

총 매출은 185000원 입니다.

ANSWER

```
americano_price = 2000
cafelatte_price = 3000
capucino_price = 3500

americanos = int(input("아메리카노 판매 개수: "))
cafelattes = int(input("카페라떼 판매 개수: "))
capucinos = int(input("카푸치노 판매 개수: "))

sales = americanos*americano_price
sales = sales + cafelattes*cafelatte_price
sales = sales + capucinos*capucino_price
print("총 매출은", sales, "입니다.")
```

• 성적계산 프로그램

이름을 입력하세요 : 파이썬

국어 성적을 입력하세요 : 95

수학 성적을 입력하세요 : 98

사회 성적을 입력하세요 : 84

과학 성적을 입력하세요 : 90

영어 성적을 입력하세요 : 79

파이썬 님의 성적은

총합 446 점, 평균 89.2 점 입니다.

ANSWER

```
name = input("이름을 입력하세요 : ")
korean = int(input("국어 성적을 입력하세요 : "))
math = int(input("수학 성적을 입력하세요 : "))
society = int(input("사회 성적을 입력하세요 : "))
science = int(input("과학 성적을 입력하세요 : "))
english = int(input("영어 성적을 입력하세요 : "))
test_sum = korean + math + society + science + english
test_average = test_sum / 5

print(name, "님의 성적은")
print("총합", test_sum, "점, 평균", test_average, "점 입니다.")
```

• 피타고라스 정리

첫번째 직각변의 길이(cm) : 15.3

두번째 직각변의 길이(cm) : 12.1

빗변의 길이는 19.50640920313116 cm입니다.

ANSWER

• 피타고라스 정리

첫번째 직각변의 길이(cm) : 15.3

두번째 직각변의 길이(cm) : 12.1

빗변의 길이는 19.50640920313116 cm입니다.

```
a = float(input("첫번째 직각변의 길이(cm) : "))  
b = float(input("두번째 직각변의 길이(cm) : "))  
print("빗변의 길이는", (a**2 + b**2)**0.5, "cm 입니다.")
```

• 원의 넓이 구하기

원의 반지름을 입력하세요(cm) : 5

원의 둘레는 31.4 cm 이고 원의 넓이는 78.5 cm입니다.

ANSWER

• 원의 넓이 구하기

원의 반지름을 입력하세요(cm) : 5

원의 둘레는 31.4 cm 이고 원의 넓이는 78.5 cm입니다.

```
r = float(input("원의 반지름을 입력하세요(cm) : "))  
rod = round(2*3.14*r, 2)  
area = round(3.14*(r**2), 2)  
print("원의 둘레는", rod, "cm 이고 원의 넓이는 ", area, "cm입니다.")
```


PRACTICE

• 근의 공식

이차방정식 $ax^2 + bx + c$ 해 계산기입니다.

a 값 : 3

b 값 : 4

c 값 : 1

이차방정식 $3x^2 + 4x + 1$ 의 해는

-4 ± 2.0

6

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-4 \pm 2.0}{6}$$

ANSWER

- 근의 공식

```
print("이차방정식  $ax^2 + bx + c$  해 계산기입니다.")
a = int(input("a 값 : "))
b = int(input("b 값 : "))
c = int(input("c 값 : "))
print("이차방정식 ", a, " $x^2 +$ ", b, " $x +$ ", c, "의 해는")
print()
print(-b, "±",  $(b^2 - 4ac)^{0.5}$ )
print("-----")
print(" ",  $2a$ , " ")
```

PRACTICE

• 화씨 온도를 섭씨로 변환하기

화씨 온도를 입력하세요 : 100

섭씨 온도는 : 37.777777777777778

$$C = (F - 32) * \frac{5}{9}$$

ANSWER

• 화씨 온도를 섭씨로 변환하기

화씨 온도를 입력하세요 : 100

섭씨 온도는 : 37.77777777777778

$$C = (F - 32) * \frac{5}{9}$$

```
ftemp = int(input("화씨 온도를 입력하세요 :"))  
ctemp = (ftemp-32.0)*5.0/9.0  
print("섭씨 온도는 :",ctemp)
```

PRACTICE

- 섭씨 온도를 화씨로 변환하기

섭씨 온도를 입력하세요 :

화씨 온도는 :

$$C = (F - 32) * \frac{5}{9}$$

• 피타고라스 정리

첫번째 직각변의 길이(cm) : 15.3

두번째 직각변의 길이(cm) : 12.1

빗변의 길이는 19.50640920313116 cm입니다.

ANSWER

• 피타고라스 정리

첫번째 직각변의 길이(cm) : 15.3

두번째 직각변의 길이(cm) : 12.1

빗변의 길이는 19.50640920313116 cm입니다.

```
a = float(input("첫번째 직각변의 길이(cm) : "))  
b = float(input("두번째 직각변의 길이(cm) : "))  
print("빗변의 길이는", (a**2 + b**2)**0.5, "cm 입니다.")
```

• 입력받은 문자 출력하기

첫번째 이름을 입력하세요 : 동팔이

두번째 이름을 입력하세요 : 진수

세번째 이름을 입력하세요 : 미친개

동팔이 님 올해에도 새해 복 많이 받으시고 만수무강 하세요.

진수 님 올해에도 새해 복 많이 받으시고 만수무강 하세요.

미친개 님 올해에도 새해 복 많이 받으시고 만수무강 하세요

ANSWER

• 입력받은 문자 출력하기

첫번째 이름을 입력하세요 : 동팔이

두번째 이름을 입력하세요 : 진수

세번째 이름을 입력하세요 : 미친개

동팔이 님 올해에도 새해 복 많이 받으시고 만수무강 하세요.

진수 님 올해에도 새해 복 많이 받으시고 만수무강 하세요.

미친개 님 올해에도 새해 복 많이 받으시고 만수무강 하세요

```
name1 = input("첫번째 이름을 입력하세요 : ")
name2 = input("두번째 이름을 입력하세요 : ")
name3 = input("세번째 이름을 입력하세요 : ")
print(name1, "님 올해에도 새해 복 많이 받으시고 만수무강 하세요.")
print(name2, "님 올해에도 새해 복 많이 받으시고 만수무강 하세요.")
print(name3, "님 올해에도 새해 복 많이 받으시고 만수무강 하세요.")
```

PRACTICE

• BMI 계산하기

이름을 입력하세요 : 파이썬

키(cm)를 입력하세요 : 176

몸무게(kg)를 입력하세요 : 72

파이썬님의 키는 176 cm이고 몸무게는 72 kg 입니다.

BMI 지수는 23.24 입니다.

$$\text{BMI} = \frac{(\text{weight in kilograms})}{\text{height in meters}^2}$$



소수점을 반올림하여 나타내는 함수는 round(숫자, 나타낼 자리 수)이다.

round(3.141592, 2) = 3.14

• BMI 계산하기

이름을 입력하세요 : 파이썬

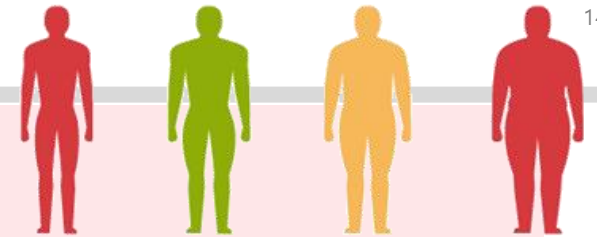
키(cm)를 입력하세요 : 176

몸무게(kg)를 입력하세요 : 72

파이썬님의 키는 176 cm이고 몸무게는 72 kg 입니다.

BMI 지수는 23.24 입니다.

$$\text{BMI} = \frac{(\text{weight in kilograms})}{\text{height in meters}^2}$$



BMI Chart

BMI less than 18.50	Underweight
BMI 18.50 - 24.99	Healthy weight
BMI 25.00 - 29.99	Overweight
BMI 30 or more	Obese

소수점을 반올림하여 나타내는 함수는 round(숫자, 나타낼 자리 수)이다.

round(3.141592, 2) = 3.14

```
name = input("이름을 입력하세요 : ")
height = int(input("키(cm)를 입력하세요 : "))
weight = int(input("몸무게(kg)를 입력하세요 : "))
bmi = weight / (height / 100)**2
print()
print(name+"님의 키는 ", height, "cm이고 몸무게는 ", weight, "kg 입니다.")
print("BMI 지수는 ", round(bmi, 2), "입니다.")
```

PRACTICE

• 자동판매기 프로그램

자동 판매기를 시뮬레이션하는 프로그램을 작성해보자. 자동 판매기는 사용자로부터 투입한 돈과 물건값을 입력받는다. 물건값은 100원 단위라고 가정한다. 프로그램은 잔돈을 계산하여 출력한다. 자판기는 동전 500원, 100원짜리만 가지고 있다고 가정하자

투입한 돈 : 5000

물건 값 : 2600

거스름 돈 : 2400

500원 동전의 개수 : 4

100원 동전의 개수 : 4

소수점을 반올림하여 나타내는 함수는 `round(숫자, 나타낼 자리수)`이다.

`round(3.141592, 2) = 3.14`

ANSWER

• 자동판매기 프로그램

```
money = int(input("투입한 돈: "))  
price = int(input("물건 값 : "))
```

```
change = money - price  
print("거스름돈: ", change)  
coin500s = change // 500  
change = change % 500  
coin100s = change // 100
```

*# 500으로 나누어서 몫이 500원짜리의 개수
500으로 나눈 나머지를 계산한다.
100으로 나누어서 몫이 100원짜리의 개수*

```
print("500원 동전의 개수 :", coin500s)  
print("100원 동전의 개수 :", coin100s)
```

- 문자열 반복

*******Hello, Python*******

다음 코드를 이용하여 위의 결과를 출력하자

```
string1 = 'Hello,'  
string2 = ' Python'  
star = '*'
```

ANSWER

- 문자열 반복

*******Hello, Python*******

다음 코드를 이용하여 위의 결과를 출력하자

```
string1 = 'Hello,'  
string2 = ' Python'  
star = '*'
```

```
string1 = 'Hello,'  
string2 = ' Python'  
star = '*'  
print(star * 9 + string1 + string2 + star*9)
```

PRACTICE

• 문자열 출력

출력 뒤 각각 해당하는 내용을 입력하여 fullname으로 출력하는 프로그램을 작성하라.

Last Name을 입력하시오: Hong

First Name을 입력하시오: Gildong

Hong Gildong

ANSWER

• 문자열 출력

출력 뒤 각각 해당하는 내용을 입력하여 fullname으로 출력하는 프로그램을 작성하라.

Last Name을 입력하시오: Hong

First Name을 입력하시오: Gildong

Hong Gildong

```
last_name = input("Last Name을 입력하시오:")  
first_name = input("First Name을 입력하시오:")  
print(last_name, first_name)
```

• INPUT

시각을 입력하시오: 3

분을 입력하시오: 30

초를 입력하시오: 15

3시 30분 15초 입니다.

ANSWER

• INPUT

시각을 입력하시오: 3

분을 입력하시오: 30

초를 입력하시오: 15

3시 30분 15초 입니다.

```
hour = input("시각을 입력하시오:")  
min = input("분을 입력하시오:")  
sec = input("초를 입력하시오:")  
print(hour+"시 "+ min +"분 "+ sec +"초")
```

- 비교연산자

숫자를 입력받아 짝수이면 True 홀수이면 False를 출력하는 프로그램을 작성하라

숫자를 입력하세요: 10

True

ANSWER

• 비교연산자

숫자를 입력받아 짝수이면 True 홀수이면 False를 출력하는 프로그램을 작성하라

숫자를 입력하세요: 10
True

```
n = int(input("숫자를 입력하세요: "))  
flag = int(n % 2)  
flag = (flag == 0)  
print("%s"%flag)
```