

# 파이썬 기본

## 2. 변수와 수식



python

# 01. 변수

- 변수 : 특정 값을 저장하는 메모리 공간

어떤 타입의 데이터도 저장 가능, 선언하지 않아도 된다.

변수이름 = 값 (할당문 또는 대입문)

변수에 있는 값은 언제든지 변경 가능

```
>>> x = 10
```

```
>>> x = 20
```

```
>>> print(x)
```

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

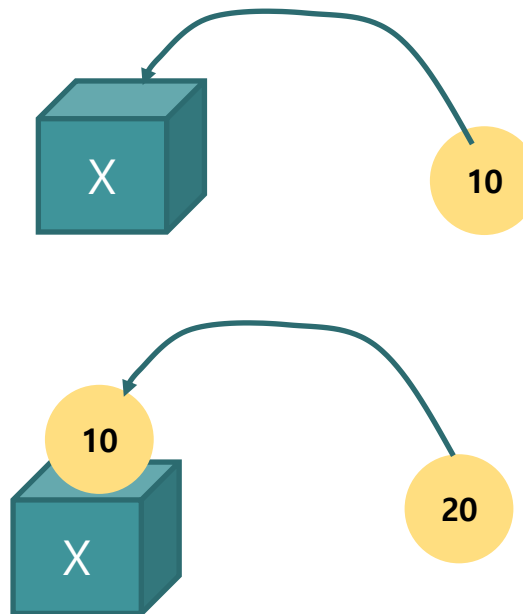
```
>>> y = 200
```

```
>>> sum = x + y
```

```
>>> print(sum)
```

```
>>> sum = sum + 100
```

```
>>> name = "홍길동"
```





## 02. 식별자 기본 규칙

- 식별자 – 이름을 붙일 때 사용하는 단어
  - 변수 이름, 함수 이름, 클래스 이름
- 변수 이름 규칙
  - 영문자와 숫자, 밑줄 문자(\_)로 이루어 진다
  - 중간에 공백이 들어가면 안된다
  - 첫 글자는 반드시 영문자 또는 밑줄 문자(\_)
  - 예약어(키워드)는 사용 불가 – if, elif, while, for...
  - 대문자와 소문자는 구별된다 - sum ≠ SUM
  - 내장함수는 사용하지 않는 것이 좋다 – print, sum, abs ...
  - 변수의 역할을 가장 잘 설명하는 이름을 지어야 한다



## 02. 식별자 기본 규칙

➤ 키워드 – 특별한 의미가 부여된 단어

False	None	True	and	as	assert
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			

```
>>> import keyword
```

```
>>> print(keyword.kwlist)
```



## 02. 식별자 기본 규칙

### ➤ 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.



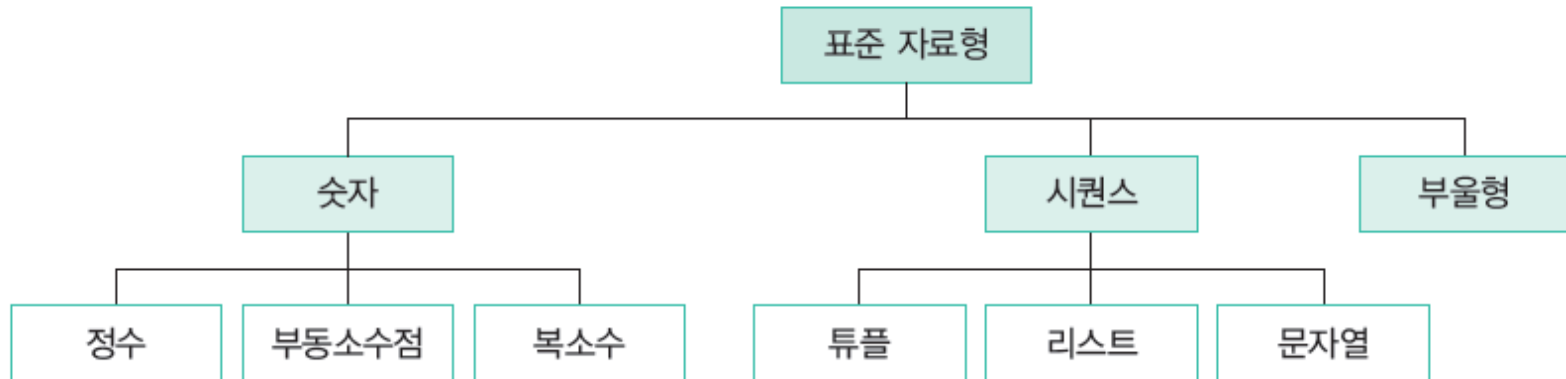
## 02. 식별자 기본 규칙

- 스네이크 케이스(snake\_case) : 단어 사이에 \_ 기호를 붙여 식별자 만든다
- 캐멀 케이스(CamelCase) : 단어들의 첫 글자를 대문자로 만들어 식별자를 만든다  
단어들의 첫 글자는 소문자로, 나머지 단어의 첫 글자는 대문자로 적는 방법(myNewCar)

스네이크 케이스	캐멀 케이스
item_list	ItemList
login_status	LoginStatus
character_hp	CharacterHp
rotate_angle	RotateAngle



## 02. 자료형



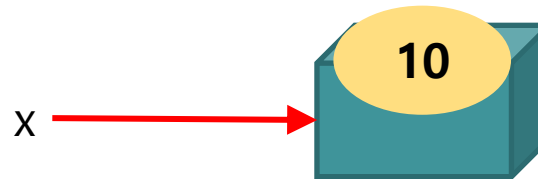
자료형	예
정수(int)	..., -2, -1, 0, 1, 2, ...
부동소수점수(float)	3.2, 3.14, 0.12
문자열(str)	'Hello World!', "123"

```
>>> type(10)
>>> type(True)
>>> type(12.30)
>>> type("hello")
```

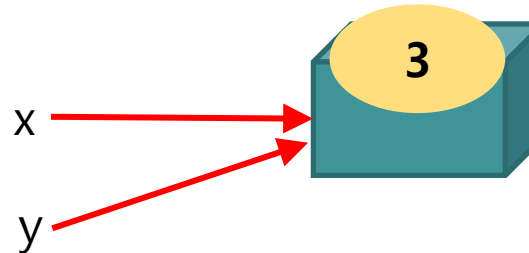
# 03. 변수의 세부 구현 사항

- 파이썬에서는 모든 것이 객체로 되어있다
- 변수에 저장되는 것은 실제 값이 아니고 객체의 참조값(주소)이다

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

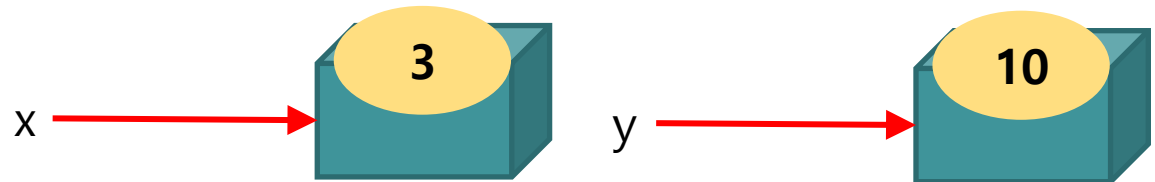


```
>>> x = 3  
>>> y = x
```



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

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





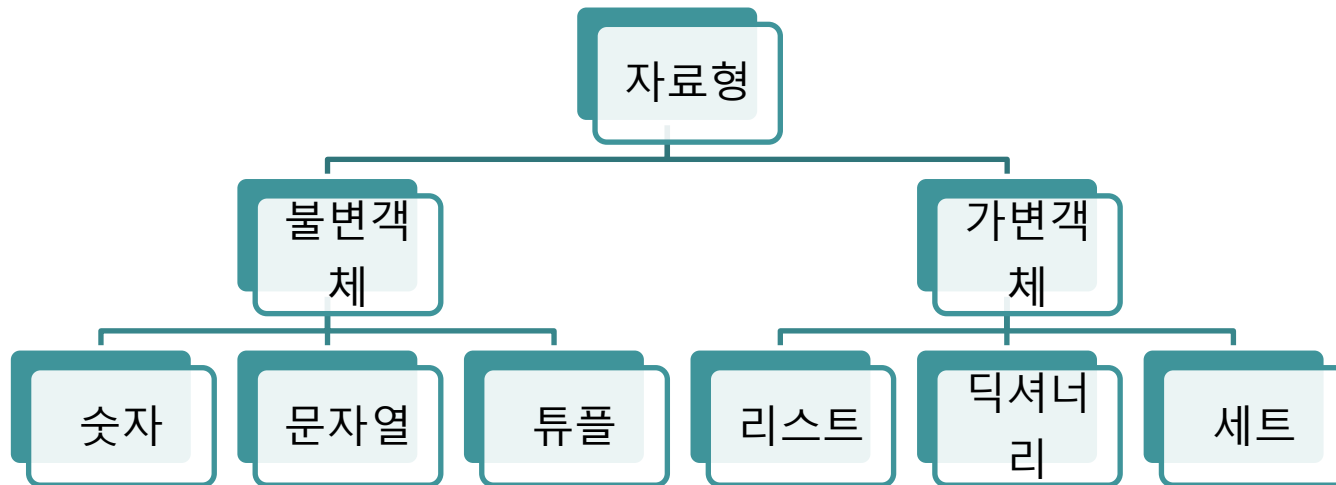
# 03. 변수의 세부 구현 사항

➤ 불변 객체 : 한번 만들어지면 변경할 수 없는 객체

정수, 실수, 문자열, 튜플

➤ 가변 객체 : 직접 변경할 수 있는 객체

리스트, 딕셔너리, 세트, ...





## 04. 주석과 상수

➤ 주석(comment) : 소스 코드에 붙이는 설명글

# 으로 시작

➤ 상수(constant) : 한번 값이 결정되면 절대로 변경되지 않는 변수

파이썬에서는 상수를 정의하지 않는다

변수 이름을 대문자로 하여 구분하는 것도 좋다

```
TAX_RATE = 0.35
```

```
PI = 3.141592
```

```
MAX_SIZE = 100
```

```
>>> TAX_RATE = 0.35
```

```
>>> tax= 1000 * TAX_RATE
```

```
>>> income = 1000 - tax
```

```
>>> print("현재의 세율은", TAX_RATE * 100, "퍼센트 입니다.")
```



# 05. 수식

## ➤ 산술 연산

연산자	기호	사용예	결과값
덧셈	+	$7 + 4$	11
뺄셈	-	$7 - 4$	3
곱셈	*	$7 * 4$	28
나눗셈	//	$7 // 4$	1
나눗셈	/	$7 / 4$	1.75
나머지	%	$7 \% 4$	3



# 05. 수식

## ➤ 할당 연산

```
>>> x = 10
```

```
>>> y = "Hello, World!"
```

```
>>> z = 3.141592
```

```
>>> type(x)
```

```
>>> type(y)
```

```
>>> type(z)
```

```
>>> a, b, c = 10, 20, 30
```

```
>>> a
```

```
>>> b
```

```
>>> c
```

```
>>> a = b = c = 100
```

```
>>> del x ➔ 변수 삭제
```



# 05. 수식

## ➤ 복합연산자

복합 연산자	의미
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$
$x %= y$	$x = x \% y$



# 05. 수식

- 연산자의 우선 순위
  - 연산자의 우선 순위 변경 ➔ ( )

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

>>> -5 \*\* 2

>>> 23 + 22 \* 5

>>> (-5) \*\* 2

>>> (23 + 22) \* 5



## 05. 수식

순위	연산자	설명
8	<= < > >=	비교 연산자
9	<> == !=	동등 연산자
10	= %= /= //= -= += *= **=	대입 연산자
11	is is not	동등 연산자
12	in not in	소속 연산자
13	not or and	논리 연산자



## 06. 자료의 종류

- 파이썬에서 사용 가능한 기본 자료형

자료형	예
정수	..., -2, -1, 0, 1, 2, ...
실수	3.2, 3.14, 0.12
문자열	'Hello World!', "123"

- 변수에 어떤 종류의 자료도 저장 가능하다.

```
>>> x = 10
>>> print("x = ", x)
```

```
>>> x = 3.14
>>> print("x = ", x)
```

```
>>> x = "Hello world!"
>>> print("x = ", x)
```





## 07. 숫자형

- 숫자 형태로 이루어진 자료형

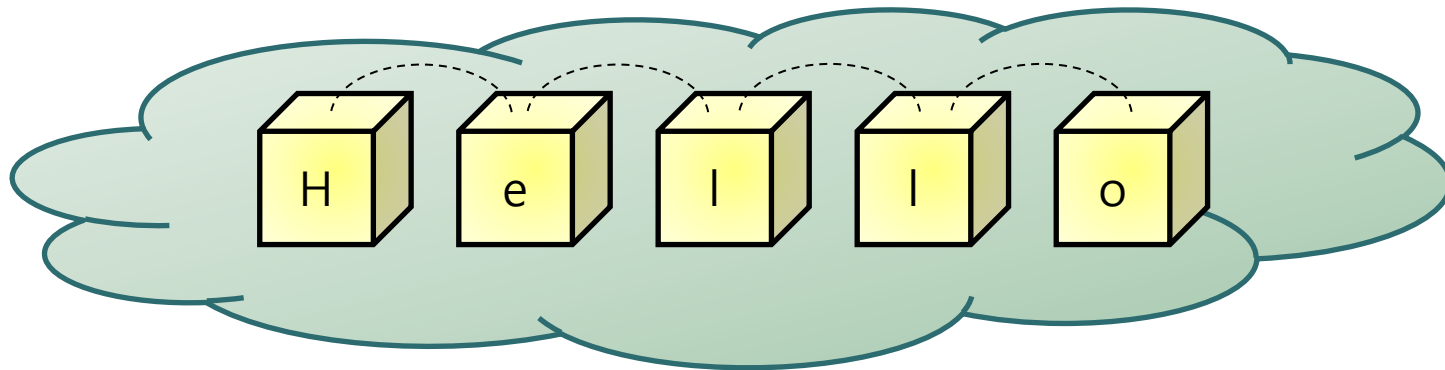
항목	파이썬 사용 예
정수	123, -345, 0
실수	123.45, -1234.5, 3.4e10, 3.4E2
8진수	0o34, 0o25
16진수	0x2A, 0xFF

```
>>> a = 123
>>> b = 1.2
>>> print(3.4e5)
>>> print(5.2e-3)
>>> print(0o34)
>>> print(0x2A)
```



## 08. 문자열

- 문자열(String) : 문자, 단어 등으로 구성된 문자들의 집합  
큰따옴표("...") 또는 작은따옴표('...')로 감싸서 표현  
큰따옴표 3개를 연속으로 써서 감싸기 ("\"\"\"...\"\"\"")  
작은따옴표 3개를 연속으로 써서 감싸기 ('\"\"\"...\"\"\"')





## 08. 문자열

```
>>> print("철수가 '안녕 ' 이라고 말했습니다.")
>>> print(100 + 200)
>>> print("100" + "200")
>>> int(3.14)
>>> int(3.9)  ➔ 소수 이하 버림
>>> float(2)  ➔ 실수로 변환
>>> str(2)    ➔ 문자열로 변환
>>> print('나는 현재 ' + 21 + '살이다') ➔ 에러
>>> print('나는 현재 ' + str(21) + '살이다')
>>> line = '=' * 50
>>> print(line)
```



## 08. 문자열

```
>>> food = "Python's favorite food is perl"
```

```
>>> food
```

```
>>> food = 'Python\'s favorite food is perl'
```

```
>>> multiline = "Life is too short\nYou need python"
```

```
>>> multiline = """
```

```
Life is too short
```

```
You need python
```

```
"""
```

```
>>> multiline = "Life is too short\n"
```

```
>>> multiline += "You need python"
```

```
>>> print(multiline)
```

```
>>> a = "Life is too short"
```

```
>>> len(a) → 문자열 길이
```



## 09. 특수 문자열

- 이스케이프 문자 - \ (백슬래시)와 같이 사용
- print() 명령에 같이 사용

이스케이프 문자	설명
\"	큰따옴표
'	작은따옴표
\n	줄 바꿈
\t	탭
\\	\



# 11. 문자열 메소드

0	1	2	3	4	5	6	7	8	9	10	11	← 인덱스
M	o	n	t	y		p	y	t	h	o	n	
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	

```
>>> s = "Monty python"
```

```
>>> print(s[1])
```

```
>>> print(s[6:10])
```

```
>>> print(s[:4])
```

```
>>> print(s[6:])
```

```
>>> print(s[:])
```



# 11. 문자열 메소드

➤ 문자열에서 문자 추출

```
>>> a = '20010331Rainy'
```

```
>>> date = a[:8]
```

```
>>> weather = a[8:]
```

```
>>> date
```

```
>>> weather
```

문제

위 데이터를 2001년 03월 31일 날씨 : Rainy 로 나타내 보세요

# 12. 입력

- input() 함수 – 입력을 무조건 문자열 형태로 반환  
    <변수> = input(<프롬프트 문자열>)

```
>>> input()
>>> name = input("name ? ")
>>> print(name)
>>> money = input("How much? ")
>>> print(money)
>>> print(money + 100)
>>> money = int(input("How much ? "))
>>> print(money)
>>> print(money + 100)
```





## 12. 입력

```
변수1, 변수2 = input().split()
```

```
변수1, 변수2 = input().split("기준문자열")
```

```
변수1, 변수2 = input("문자열").split()
```

```
변수1, 변수2 = input("문자열").split("기준문자열")
```

```
>>> a, b = input("숫자 두 개를 입력하세요 : ").split()
```

```
23 45
```

```
>>> print(a)
```

```
>>> print(b)
```

```
>>> a, b = input("숫자 두 개를 입력하세요 : ").split(',')
```

```
34,55
```

```
>>> print(a)
```

```
>>> print(b)
```



# 13. 출력

➤ print() 함수

print(<출력하고 싶은 것>,...)

```
>>> print("Hello")
```

```
>>> print(100)
```

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

```
>>> print("Good", "morning!")
```

```
>>> print(10, 20, 30)
```

```
>>> print(1000, "원")
```

```
>>> print("Good" + "morning!")
```

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

```
>>> y = 200
```

```
>>> sum = x + y
```

```
>>> print(x, "과", y, "의 합은", sum, "입니다")
```



# 13. 출력

```
>>> print("Hello")
>>> print('Hello')
>>> print("""안녕하세요""라고 말했습니다")
>>> print("""안녕하세요""라고 말했습니다')
>>> print("₩안녕하세요₩라고 말했습니다)
>>> print("say₩nhello")
>>> print("name₩tage₩tcity")
>>> print("kim₩t25₩tseoul)
>>> print("₩₩ ₩₩ ₩₩ ₩₩")
```

➤ """ 문자열 """, ' ' 문자열 ' ' ➔ 여러 줄 문자열



# 14. 형식화된 출력

- 문자열 안의 특정한 값을 바꿀 수 있게 한다

예) 현재 온도는 18도 입니다

현재 온도는 20도 입니다

문자열 포맷 코드

코드	설명
%s	문자열(String)
%c	문자 1개(Character)
%d	정수(Integer)
%f	부동 소수(Floating-point)
%o	8진수
%x	16진수
%%	Literal % (문자 '%' 자체)



# 14. 형식화된 출력

>>> "I eat %d apples." %3 → 숫자

>>> "I eat %s apples." %five → 문자열

>>> number = 3

>>> "I eat %d apples." %number → 변수

>>> number = 3

>>> day = "three"

>>> "I ate %d apples, so I was sick for %s days." %(number, day)

→ 2개 이상의 값 넣기

>>> "I have %s apples" %3 → '3'을 문자열로 처리

>>> "rate is %s" %3.234 → '3.234'를 문자열로 처리

Error is 98%.

>>> "Error is %d%%" %98



# 14. 형식화된 출력

## ➤ 정렬과 공백

```
>>> "%10s" % "Hi"
```

```
'      Hi'
```

```
>>> "%-10s" % 'Hi'
```

```
'Hi      '
```

```
>>> "%-10sjane" % 'Hi'
```

```
'Hi      jane'
```

## ➤ 소수점 표현하기

```
>>> "%0.4f" % 3.141592
```

```
>>> "%10.4f" % 3.141592
```



# 14. 형식화된 출력

➤ format 함수를 사용한 포매팅

```
>>> "I eat {0} apples.".format(3)
```

```
>>> "I eat {0} apples.".format("five")
```

```
>>> number = 3
```

```
>>> "I eat {0} apples.".format(number)
```

```
>>> number = 10
```

```
>>> day = "three"
```

```
>>> "I ate {0} apples, so I was sick for {1} days".format(number, day)
```

```
>>> "I ate {number} apples, so I was sick for {day}
```

```
days".format(number=10, day=3)
```

```
>>> "I ate {0} apples, so I was sick for {day} days".format(10, day=3)
```



# 14. 형식화된 출력

➤ format 함수를 사용한 포매팅 - 정렬

```
>>> "{0:<10}".format("hi")
```

```
'hi          '
```

```
>>> "{0:>10}".format("hi")
```

```
'          hi'
```

```
>>> "{0:^10}".format("hi")
```

```
'   hi   '
```

```
>>> "{0:=^10}".format("hi")
```

```
'====hi===='
```

```
>>> "{0:!  
<10}".format("hi")
```

```
'hi!!!!!!!!'
```





# 실습문제

1. 두수 50 과 100 의 사칙연산을 + , - , \* , / (산술연산자) 를 이용해 구하고 연산결과 값을 화면에 출력하세요
  2. 첫번째 변수에 'python is' 를 저장 하고 두번째 변수에 'easy' 를 사용자에게 입력 받아 'python is easy' 를 화면에 출력하세요
  3. 사각형의 면적을 구하려고 합니다. 가로와 세로의 길이를 입력 받아서 사각형의 넓이를 계산하는 프로그램을 작성하세요
- 예) 사각형의 가로의 길이 : 30  
사각형의 세로의 길이 : 20  
사각형의 넓이 : 600



# 실습문제

4. 초단위의 시간을 입력 받아서 몇 분 몇 초인지 계산하여 출력하세요

예) 초단위의 시간을 입력하세요 : 1000

1000초는 16분 40초 입니다.

5. 가지고 있는 돈으로 가격이 120원인 사탕을 최대한 몇 개까지 살 수 있는지 계산하고, 나머지 돈은 얼마인지 나타내시오

예) 가지고 있는 돈은 얼마입니까? 5000

살 수 있는 사탕의 수 : 41 개

사탕을 구입하고 남은 돈 : 80 원



# 실습문제

6. input () 함수로 두 수를 입력 받는다.

input() 함수의 도움말 인수는 'number1 : ' 과 'number2 : ' 로 한다.

입력 받은 두수의 + , - , \* , / 의 연산 결과를 화면에 출력한다  
나눗셈의 결과는 소수 둘째자리까지 나타낸다.

예) number1 : 23

number2 : 3

$23 + 3 = 26$

$23 - 3 = 20$

$23 \times 3 = 69$

$23 / 3 = 7.67$



# 실습문제

## 7. 자동판매기 프로그램

사용자는 1000원 짜리 지폐와 500원 짜리 동전, 100원 짜리 동전을 사용할 수 있다. 물건값을 입력하고 1000원권, 500원 짜리 동전, 100원 짜리 동전의 개수를 입력하면 거스름돈을 계산하여 동전으로 반환한다.

예) 물건값을 입력하세요 : 750

1000원 지폐개수 : 1

500원 동전개수 : 0

100원 동전개수 : 0

거스름돈

500원 = 0    100원 = 2    10원 = 5