

IT 개론

3장. 입출력문과 연산자

목차

1. 입출력문 (print, input)
2. 산술 연산자
3. 관계 연산자
4. 논리 연산자
5. 수치 연산 함수들

1. 입출력문

◆ 출력문 – print() 내장 함수

함수는 반드시 괄호를
사용한다.

Python2

```
>>> print "hello"
hello
>>> print 'hello'
hello

>>> print("hello")
hello
>>> print('hello')
hello
```

Python2에서는 괄호를
써도 되고 안 써도 된다.

Python3 : print 함수로만 사용 가능

```
>>> print "hello"
SyntaxError: Missing parentheses .....
>>> print 'hello'
SyntaxError: Missing parentheses .....

>>> print("hello")
hello
>>> print('hello')
hello
```

Python3에서는 반드시 괄호를 써야 한다.

1. 입출력문

◆ 출력문 – print() 내장 함수

- print() 함수는 괄호의 내용을 출력한다.
- 홑따옴표('...') 또는 쌍따옴표("...")를 이용하여 원하는 문자열을 출력할 수 있다.
- 홑따옴표 세 개(''...'') 또는 쌍따옴표 세 개('"'...'")를 사용할 수도 있다.
- 여러 값을 콤마로 구분할 수 있으며, 출력할 때 각각의 값 사이에 공백 한 개가 추가된다.

1. 입출력문

◆ 출력문 – print() 내장 함수

```
>>> print("hello", "world") # 콤마로 분리할 때 스페이스가 추가된다.
```

```
hello world
```

```
>>> print('hello', 'world')
```

```
hello world
```

```
>>> a = 10
```

```
>>> print('a =', a)
```

```
a = 10
```

```
>>> print(m) # 따옴표가 없으면 변수로 간주한다.
```

```
Traceback (most recent call last):
```

```
  File "<pysHELL#56>", line 1, in <module>
```

```
    print(m)
```

```
NameError: name 'm' is not defined
```

```
>>> print('m')
```

```
m
```

1. 입출력문

◆ 출력문 - print() 내장 함수와 특수 문자

| 특수문자 | 설명 |
|------|-----------|
| \ | 다음 줄과 연속임 |
| \\ | \ 문자 자체 |
| \' | ' |
| \" | " |
| \n | 줄 바꾸기 |
| \t | 탭 키 |

‘\’는 키보드에서
₩ 을 누르면 된다.

```
>>> print('hello world')
hello world
>>> print('hello \ # 다음 줄과 연결됨
world')
hello world
>>> print('hello \'world\' !!')
hello 'world' !!
>>> print("hello \"world\" !!")
hello "world" !!
>>> print('hello "world"')
hello "world"
>>> print("hello 'world'")
hello 'world'
```

1. 입출력문

◆ 출력문 – print() 내장 함수와 특수 문자

```
>>> print('hello \\ world')
```

```
hello \ world
```

```
>>> print('hello \n world \n python \t programming')
```

```
hello
```

```
world
```

```
python          programming
```

1. 입출력문

◆ 출력문 - % 이용한 서식 출력

| |
|----------|
| 문자열 : %s |
| 정수 : %d |
| 실수 : %f |

```
>>> name = 'Alice'
>>> score = 95
>>> print('%s got %d score' % (name, score))
Alice got 95 score
```

```
>>> math = 93.5
>>> eng = 88.3
>>> print('Math is %5.2f and Eng is %6.3f' % (math, eng))
Math is 93.50 and Eng is 88.300
```

```
>>> print("score is %d and gpa is %5.2f" % (85, 3.7))
score is 85 and gpa is 3.70
```


1. 입출력문

◆ 입력문 – input() 내장 함수

| | | |
|---------|-------------|---|
| Python2 | input() | 데이터를 입력할 때 타입을 정확히 판단할 수 있는 형태로 넣어야 함. |
| | raw_input() | 입력되는 데이터는 모두 문자열로 취급함. |
| Python3 | input() | Python2의 input() 함수는 없어지고 raw_input() 함수의 이름이 input()으로 바뀐 것. 즉, 입력되는 데이터는 모두 문자열로 취급함. |

1. 입출력문

◆ 입력문 – input() 내장 함수

- 키보드로부터 입력을 받는다.

```
>>> x = input('Enter x : ')
Enter x : 10      # 10이 변수 x 에 저장된다.
>>> print(x)
10
>>> x
'10'
>>> type(x)      # 입력받은 데이터는 항상 문자열로 처리한다.
<class 'str'>
>>> x = int(x)    # x를 숫자로 사용하려면 int 함수 이용한다.
>>> type(x)
<class 'int'>
```

1. 입출력문

◆ 입력문 – input() 내장 함수

- 일반적으로 다음과 같이 이용한다.

```
>>> x = int(input('Enter one integer : '))
Enter one integer : 100
>>> type(x)
<class 'int'>
```

```
>>> y = float(input('Enter one float number : '))
Enter one float number : 3.14
>>> type(y)
<class 'float'>
```

7. 형변환

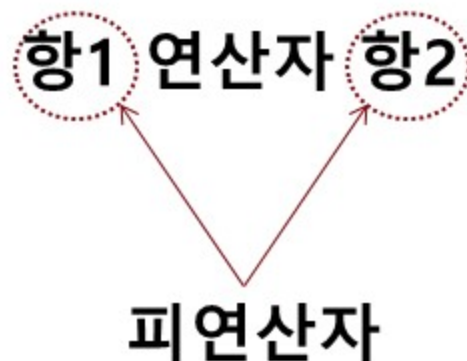
◆ 서로 다른 자료형으로 변환이 가능하다.

| 내장 함수 | 설 명 |
|------------------------|---|
| <code>int()</code> | <code>float</code> , <code>bool</code> , <code>str</code> 데이터를 <code>int</code> 형으로 변환한다. |
| <code>float()</code> | <code>int</code> , <code>bool</code> , <code>str</code> 데이터를 <code>int</code> 형으로 변환한다. |
| <code>bool()</code> | <code>int(0)</code> , <code>float(0.0)</code> , 빈 문자열, 빈 리스트, 빈 튜플, 빈 집합, 빈 사전은 모두 <code>False</code> 로 인식한다. 나머지는 모두 <code>True</code> 로 간주한다. |
| <code>complex()</code> | <code>int</code> , <code>float</code> , <code>bool</code> 을 <code>complex</code> 형으로 변환한다. |
| <code>str()</code> | <pre>>>> a = 'hello' >>> b = list(a) >>> c = tuple(a) >>> d = set(a) >>> print(b) ['h', 'e', 'l', 'l', 'o'] >>> print(c) ('h', 'e', 'l', 'l', 'o') >>> print(d) {'h', 'e', 'o', 'l'}</pre> |
| <code>list()</code> | |
| <code>tuple()</code> | |
| <code>set()</code> | |
| <code>dict()</code> | |

2. 산술 연산자

◆ 산술 연산자

| 연산자 | 의미 | 예 | 결과 |
|-----|------|---------|------|
| + | 더하기 | 10 + 5 | 15 |
| - | 빼기 | 20 - 13 | 7 |
| * | 곱하기 | 3 * 10 | 30 |
| / | 나누기 | 100 / 8 | 12.5 |
| ** | 지수계산 | 2 ** 5 | 32 |
| // | 몫 | 30 // 7 | 4 |
| % | 나머지 | 30 % 7 | 2 |



2. 산술 연산자

◆ 산술 연산자 예제

```
>>> 50 + 10
60
>>> 100 - 80
20
>>> 25 * 8
200
>>> 25 / 4
6.25
>>> 3 ** 4
81
>>> 100 // 6
16
>>> 100 % 6
4
```

[illegible]

2. 산술 연산자

◆ 산술 연산자 간략히 쓰기

| | |
|--------------|--------------------|
| $a = a + b$ | $a \text{ += } b$ |
| $a = a - b$ | $a \text{ -= } b$ |
| $a = a * b$ | $a \text{ *= } b$ |
| $a = a / b$ | $a \text{ /= } b$ |
| $a = a ** b$ | $a \text{ **=} b$ |
| $a = a // b$ | $a \text{ //= } b$ |
| $a = a \% b$ | $a \text{ \%=} b$ |

2. 산술 연산자

◆ 산술 연산자 우선순위

| 연산자 | 설명 | 결합 순서 |
|----------|------------------|-------|
| + - | 부호 | ← |
| ** | 지수 | ← |
| * / % // | 곱하기, 나누기, 나머지, 몫 | → |
| + - | 더하기, 빼기 | → |

```
>>> +3 + -5
```

```
-2
```

```
>>> 2 + 3 * 5
```

```
17
```

```
>>> 2 ** 3 ** 2
```

```
512
```

```
>>> (2 ** 3) ** 2
```

```
64
```


3. 관계 연산자

◆ 관계 연산자 - 데이터의 대소 비교 (결과 True/False)

| 관계 연산자 | 설명 |
|--------|--------|
| > | 크다 |
| >= | 크거나 같다 |
| < | 작다 |
| <= | 작거나 같다 |
| == | 같다 |
| != | 같지 않다 |

```
>>> a = 10
>>> b = 15
>>> c = 5
>>> a < b
True
>>> a != c
True
>>> a == b
False
>>> c < a < b
True
>>> c < b < a
False
```

4. 논리 연산자

- ◆ 논리 연산자는 True 또는 False를 피연산자로 취해서 True 또는 False를 결과로 낸다.

| 논리 연산자 (우선순위순) | 설명 |
|----------------|--|
| not x | x가 False이면 True가 되고, True이면 False가 된다. |
| x and y | x와 y가 모두 True인 경우만 True가 된다. |
| x or y | x와 y가 모두 False인 경우만 False가 된다. |

```
>>> a = 10; b = 20; c = 5
>>> a < b and b > c
True
>>> a < b or b < c
True
>>> a == b or b != c
True
```

```
>>> a = 10; b = 20; c = 30
>>> a < b < c
True
>>> a < b and b < c
True
```

5. 수치 연산 함수들

◆ 내장 수치 연산 함수

| 함수 | 설명 |
|--------------------------|-----------------------|
| <code>abs(x)</code> | x 의 절대값을 내준다 |
| <code>divmod(x,y)</code> | $(x//y, x\%y)$ 쌍을 구한다 |
| <code>pow(x,y)</code> | x^y 을 구한다 |

```
>>> abs(-3)
```

```
3
```

```
>>> divmod(17,4)
```

```
(4, 1)
```

```
>>> pow(2,5)
```

```
32
```

5. 수치 연산 함수들

◆ math 모듈 메소드들

```
>>> import math
>>> math.fabs(-3)          # 절대값
3.0
>>> math.factorial(5)      # 5*4*3*2*1을 계산한다
120
>>> math.pow(2,5)
32.0
>>> math.sqrt(9)
3.0
>>> math.trunc(10.9)       # 소수점 아래를 버린다
10
>>> math.log2(1024)
10.0
>>> math.log10(1024)
3.010299956639812
```