

기초 PYTHON 프로그래밍

5. 파이썬 입출력문

1. print() 함수
2. 특수 문자 출력
3. % 이용한 서식 출력
4. print() 함수와 end, sep
5. input() 함수

1. print() 함수

◆ 출력문 - print() 내장 함수

Python2

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

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

Python3 : print 함수로만 사용 가능

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

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

- Python2에서는 괄호를 써도 되고 안 써도 된다.
- Python3에서는 반드시 괄호를 써야 한다.

1. print() 함수

◆ 출력문 - print() 내장 함수

- print() 함수는 괄호의 내용을 출력한다.
- 출력하고자 하는 값이 여러 개인 경우에는 콤마로 구분할 수 있으며, 출력할 때 각각의 값 사이에 공백 한 개가 추가된다.
- 문자열을 출력하려면 홑따옴표('...') 또는 쌍따옴표("...")를 이용한다.
(홑따옴표 세 개(''... '''), 쌍따옴표 세 개('"'... '"')) 사용 가능)

```
>>> x = 10; y = 5; z = 7
>>> print(x,y,z)
10 5 7
>>> print('x')
x
>>> print(x)
10
```

1. print() 함수

◆ 변수에 저장된 값 출력하기

```
>>> a = 100 ; b = 200
```

```
>>> c = a + b
```

```
>>> print(a,b,c)  # 콤마에 공백이 추가된다.
```

```
100 200 300
```

```
>>> print(a+b)    # 수치 연산자는 계산 결과를 보여준다.
```

```
300
```

```
>>> print(a+50)
```

```
150
```

1. print() 함수

◆ 문자열 출력하기

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

```
>>> print('hello', 'world!') # 콤마에 공백이 추가된다.  
hello world!
```

```
>>> print('hello' + 'world!')  
helloworld!
```

```
>>> movie = 'toy story'
```

```
>>> print(movie)  
toy story
```

```
>>> movie  
'toy story'
```

```
>>> x = '5'  
>>> y = 5  
>>> print(x)  
5  
>>> print(y)  
5  
>>> x  
'5'  
>>> y  
5
```

1. print() 함수

◆ + 연산자

```
>>> print('hello' + 100)  # 에러
```

```
>>> print('hello' + str(100))
```

```
hello100
```

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

```
>>> n = 100
```

```
>>> n + x  # 에러
```

```
>>> n + int(x)
```

```
110
```

2. 특수 문자 출력

◆ print() 함수 내에 사용하는 특수 문자 (escape sequence)

특수문자	설명
\	다음 줄과 연속임
'\'	'
'\"'	"
\\	\ 문자 자체
\n	개행문자(줄 바꿈)
\t	탭 키

'\''는 키보드에서 \ 을 누르면 된다.

```
>>> 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'
```

2. 특수 문자 출력

◆ print() 함수 내에 사용하는 특수 문자

```
>>> print('hello ₩₩ world')
```

```
hello ₩ world
```

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

```
hello
```

```
world
```

```
python          programming
```


3. % 이용한 서식 출력

◆ 출력문 - % 이용한 서식 출력 (문자열, 정수)

문자열 : %s
정수 : %d
실수 : %f

print(' ' % (,))

```
>>> name = 'Alice' ; score = 95
```

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

Alice got 95 score

```
>>> print('%10s got %5d score' % (name, score))
```

Alice got 90 score

(10칸 잡아서 Alice 출력, 5칸 잡아서 90 출력)

3. % 이용한 서식 출력

◆ 출력문 - % 이용한 서식 출력 (실수)

```
>>> math_score = 87.3
```

```
>>> print('You got %f in math' % (math_score))
```

```
You got 87.300000 in math
```

데이터가 한 개인 경우에는
괄호가 없어도 된다

%a.bf 소수점을 포함하여 a칸을 잡고 소수점 아래 b자리까지 출력

```
>>> 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 %10.2f" % (85, 3.7))
```

```
score is 85 and gpa is 3.70
```

%.2f - 소수점 둘째자리까지 출력

4. print() 함수와 end, sep

- ◆ print() 함수는 항상 '\n'을 추가한다.

```
a = 'hello'  
b = 'world'  
print(a)  
print(b)
```

```
hello  
world
```

```
a = 'hello\n'  
b = 'world'  
print(a)  
print(b)
```

```
hello  
  
world
```

- ◆ print() 함수에 '\n' 대신에 다른 문자를 출력하고자 한다면 **end = '...'** 를 추가한다.

```
print(a, end=' '  
print(b)  
  
print(a, end='*****')  
print(b)  
  
print(a, end='\t')  
print(b)
```

```
hello world  
hello*****world  
hello    world
```

4. print() 함수와 end, sep

- ◆ 여러 데이터를 출력하고자 할 때에는 자동으로 space가 들어간다. 이 space를 다른 문자로 변경하려면 **sep**를 사용한다.

```
a = 10  
b = 20  
c = 30  
print(a,b,c)  
print(a,b,c,sep=' : ')
```

```
10 20 30  
10 : 20 : 30  
10-----20-----30  
*****  
10+++20^^^30
```

5. input() 함수

◆ 입력문 - input() 내장 함수

하나의 정수를 입력하시오 : 1000
정수 1000 을 입력하였습니다.

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() 함수가 input()으로 바뀐 것. 즉, 입력되는 데이터는 모두 문자열로 취급함.

5. input() 함수

◆ 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'>
```

5. input() 함수

◆ 입력문 - input() 내장 함수

- 필요하다면 입력 받은 데이터의 자료형을 적절히 변환해야 한다.

```
>>> x = input('정수를 입력하시오 : ')
```

```
정수를 입력하시오 : 15
```

```
>>> x + 10
```

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

```
File "<pyshell#21>", line 1, in <module>
```

```
x + 10
```

```
TypeError: Can't convert 'int' object to str implicitly
```

```
>>> int(x) + 10
```

```
25
```

5. input() 함수

◆ 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'>
```