

입력, 형식 지정 출력

중앙대학교 소프트웨어학부 신재진

키보드 입력 : input()

- 함수 호출 형식 : input([prompt])
- 기능 : prompt 인자가 있으면, 끝에 개행 문자를 붙이지 않고 표준 출력. 그런 다음 함수는 입력에서 한 줄을 읽고, 문자열로 변환해서 (줄 끝의 줄 바꿈 문자를 제거) 반환.

```
age = int( input('당신의 나이를 입력하십시오 : ') )
name, age = input('Enter your name and age : ').split()
#
num1, num2, num3 = input('숫자 3개를 입력하십시오: ').split()
num1, num2, num3 = int(num1), int(num2), int(num3)
#
num1, num2, num3 = map( int, input('숫자 3개를 입력하십시오: ').split() )
```

```
당신의 나이를 입력하십시오 : 23
Enter your name and age : Hong 23
숫자 3개를 입력하십시오: 1 2 3
숫자 3개를 입력하십시오: 1 2 3
```

Python에서 형식지정 출력방법

- 포맷 문자열 리터럴 (Formatted String Literals): 포맷 문자열 리터럴 (f-문자열) : 문자열에 f 또는 F 접두어를 붙이고 표현식을 {expression}로 작성, 문자열에 파이썬 표현식의 값을 삽입
- 문자열 format() 메서드: 중괄호와 그 안에 있는 문자들 (포맷 필드) 은 str.format() 메서드로 전달된 객체들로 치환. 중괄호 안의 숫자는 str.format() 메서드로 전달된 객체들의 위치를 가리키는데 사용
- printf-스타일 형식지정: % 연산자로 C언어의 printf와 비슷한, 문자열 포매팅. 'string' % (values)가 주어지면, string에 있는 % 인스턴스는 values 요소로 대체.

```
>>> i = 11
>>> print(f'i={i:6d} i**3={i*i*i:6d} i**2={i*i:6d}') # f-문자열
i=    11 i**3=   1331 i**2=    121
>>> print('i={0:6d} i**3={2:6d} i**2={1:6d}'.format(i, i*i, i*i*i)) # format메서드
i=    11 i**3=   1331 i**2=    121
>>> print('i=%6d i**3=%6d i**2=%6d'%(i, i*i*i, i*i)) # printf-스타일
i=    11 i**3=   1331 i**2=    121
```

Python에서 형식지정 출력방법

```
>>> i = 11
>>> print(f'i={i:<6d} i**3={i*i*i:^6d} i**2={i*i:>6d}') # f-문자열
i=11      i**3= 1331 i**2=   121
>>> print('i={0:<6d} i**3={2:^6d} i**2={1:>6d}'.format(i, i*i, i*i*i)) # format메서드
i=11      i**3= 1331 i**2=   121
>>> print('i=%-6d i**3=%6d i**2=%6d'%(i, i*i*i, i*i)) # printf-스타일
i=11      i**3=  1331 i**2=   121
```

□ printf-스타일 형식 지정

- ▣ %[flags][width][.precision][lengthmodifier][conversiontype]
- ▣ flags = '#', '0', '-', ' ', '+'
- ▣ lengthmodifier = 'h', 'l', 'L' 그러나, %ld와 %lf는 각각 %d와 %f와 동일
- ▣ conversiontype = 'd', 'i', 'o', 'x', 'e', 'f', 'g', 'c', 's', 'a' 등

Python 형식지정 미니 언어

- f-문자열, 문자열 format() 메서드에서 사용
- 표준 형식 지정자

```
format_spec ::=
[[fill]align][sign][#][0][width][grouping_option][.precision][type]
fill        ::= <any character>
align       ::= "<" | ">" | "=" | "^"
sign        ::= "+" | "-" | " "
width       ::= digit+
grouping_option ::= "_" | ","
precision   ::= digit+
type        ::= "b" | "c" | "d" | "e" | "E" | "f" | "F" |
"g" | "G" | "n" | "o" | "s" | "x" | "X" | "%"
```

- printf-스타일 형식 지정 % 대신에 {}와 : 를 사용. 예) '%03.2f'는 '{:03.2f}'