

입력 함수 input()

- 키보드로부터 문자열을 입력 받는 함수

```
variable_name = input("Prompt string")
```

- ① input() 함수의 인수로 지정된 문자열이 출력(prompt)됨
- ② 그 문자열 뒤에 커서가 깜박이며 사용자로부터의 입력 기다림
- ③ 사용자가 키보드로 값을 입력하고 Enter 키를 누르면
- ④ 입력 받은 값이 문자열로 input() 함수의 반환 값이 됨
- ⑤ 반환된 값은 variable_name에 할당되어 사용

```
>>> name = input("Enter your name:")
Enter your name:Eric
>>> name
'Eric'
>>>
```

사용자가 입력한 데이터

prompt

input() 함수의 반환값의 데이터형은 string!!!
input() 함수를 통하여 string이 아닌 다른 데이터형을 입력 받으려면,
입력 받은 string형을 다른 데이터형으로 변환하여야 함

입력 함수 input()

- 입력 데이터가 두 개인 경우
 - 문자열 method인 **split()**을 사용하여 분리함
 - input() 함수로 부터 입력 받은 문자열을 문자열의 메소드 split() 을 사용하여 2개의 문자열로 분리
 - 입력 받는 데이터의 개수를 정확히 알고 있는 경우 사용 가능한 방법임

```
>>> n = input("Enter two names : ")  
Enter two names : Thom Bob  
>>> n1, n2 = n.split()  
>>> print(n1, n2) # type of n1 and n2 is string  
Thom Bob
```

입력받는 데이터가 공백으로 구분되어 있는 경우

```
>>> n1, n2, n3 = input("Enter three names : ").split()  
Enter three names : Thom Bob nill  
>>> print(n1, n2, n3) # type of n1, n2 and n3 is string  
Thom Bob nill
```

입력 함수 input()

- input() 함수는 입력 받은 데이터를 문자열로 반환함
- 다른 데이터 형을 입력 받기 위해서는 입력된 문자열을 원하는 데이터 형으로 변환해야 함

```
>>> a = int ( input("Enter an integer : ") )
```

```
Enter an integer : 123
```

```
>>> b = input("Enter a float : ")
```

```
Enter a float : 3.14
```

```
>>> b = float(b)
```

```
>>> c = input("Enter a string : ")
```

```
Enter a string : Hello
```

```
>>> print(a, b, c)
```

```
123 3.14 Hello
```

```
>>> print(type(a), type(b), type(c))
```

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

입력 함수 input()

- 두 개의 숫자 값을 입력 받는 경우
- input() 함수에서 입력 받은 문자열 데이터를 두개의 데이터로 분리한 다음, 각 각 원하는 다른 데이터 형으로 변환해야 함

```
>>> nums = input("Enter two numbers : ")
Enter two numbers : 10 20
>>> num1, num2 = nums.split() # type of n1 and n2 is string
>>> print(type(num1), type(num2))
<class 'str'> <class 'str'>
>>> num1 = int(num1); num2 = int(num2) # type of n1 and n2 is int
>>> print(type(num1), type(num2))
<class 'int'> <class 'int'>
```

```
>>> nums = int(input("Enter two numbers : "))
Enter two numbers : 10 20
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>  nu
    ms = int(input("Enter two numbers : "))
ValueError: invalid literal for int() with base 10: '10 20'
```

출력 함수 print()

- 모니터 화면에 결과를 출력하는 함수
- 함수의 인수를 모두 문자열로 바꾸어 출력하며, 함수의 인수들은 출력하고자 하는 객체들임

```
>>> a = 5; b = 10.5
>>> print("a =", a, "b =", b)
a = _5_ b = _10.5
>>> print("a = ", a, ", b = ", b)
a = _5_, b = _10.5
```

- 함수의 인수, 객체와 객체 사이에는 기본값으로 빈칸이 추가되어 출력 됨(위의 붉은 밑줄)

출력 함수 print()

- print() 함수는 출력시 기본 값으로 출력 문자열 마지막에 **Wn**(줄 바꿈 문자)가 더해짐. 즉 출력시 줄바꿈이 발생함

```
>>> print("First printing."); print("Second printing.")
```

```
First printing. S 줄바꿈 발생  
econd printing.
```

```
>>>
```

- 줄 바꿈이 필요 없는 경우, print() 함수의 **end 인자** 설정을 바꿔주면 됨

```
>>> print("First printing.", end = ' '); print("Second printing.")
```

```
First printing. Second printing.
```

```
>>> print("First printing.", end = '#'); print("Second printing.")
```

```
First printing.#Second printing.
```

```
>>> print("First printing.", end = ''); print("Second printing.")
```

```
First printing.Second printing.
```

```
>>>
```

줄바꿈 대신, 지정한 문자가 문자열의 마지막에 첨가

빈 문자열로 지정