

Simple Input/Output

By 윤명근 / 박수현

수업목표

- `input()`
- `print`
- Inputting numbers
- Input from the Web

input()


- input 함수
 - 명령 prompt에서 사용자로부터 입력 받는 함수
 - 사용자가 무엇을 입력해도 함수의 결과 값은 문자열(string) 자료형

<입력 받을 변수> = input(<프롬프트 문자열>)

```
>>>  
>>> string = input("입력> ")  
입력> Park  
>>>  
>>> print(string)  
Park
```

input()

- 예제 1

 3_input_test1.py

File Edit Format Run Options Window Help

```
# 이름을 입력받음
somebody = input("-- 이름을 입력하세요 : ")
print("1) Hi, " + somebody + "님")

# 나이를 입력받음
age = input("-- Enter your age : ")
print ("2) Your age is = ", age, "(String type)")
```

```
-- 이름을 입력하세요 : 섬바디
1) Hi, 섬바디님
-- Enter your age : 20
2) Your age is = 20 (String type)
>>>
```

input()

```
3_input_test2.py
File Edit Format Run Options Window Help
# 이름을 입력받음
somebody = input("-- Enter your name : " )
print("1) Hi, " + somebody + "님")

# 나이를 입력받음
age = input("-- Enter your age : " )
print ("2) Your age is = ", age, "(String type)")

print ("3) 10 years later, ", somebody, " will be ", age + 10)
print ("4) Bye !")
```

```
-- Enter your name : 강아지
1) Hi, 강아지님
-- Enter your age : 5
2) Your age is = 5 (String type)
Traceback (most recent call last):
  File "C:\소사\강의예제\3_input_test2.py", line 9, in <module>
    print ("3) 10 years later, ", somebody, " will be ", age + 10)
TypeError: can only concatenate str (not "int") to str
>>>
```

- 예제 2
 - 사용자 입력 String의 숫자 변환
 - String + 숫자 → error 발생

input()

- 예제 3
 - 사용자 입력 String을 숫자로 변환 (type casting)
 - int(), float() 사용

3_input_test3.py

File Edit Format Run Options Window Help

이름을 입력받음

somebody = input("-- Enter your name : ")

print("1) Hi, " + somebody + "님")

나이를 입력받음

age = int(input("-- Enter your age : "))

print ("2) Your age is = ", age, "(Int type)")

print ("3) 10 years later, ", somebody, " will be ", age+10)

print ("4) Bye !")

-- Enter your name : Park

1) Hi, Park님

-- Enter your age : 25

2) Your age is = 25 (Int type)

3) 10 years later, Park will be 35

4) Bye !

String Concatenation

- '+' 연산
 - String 연결 (string concatenation)
 - 숫자 합산 (add)

```
>>> good = "Good"
>>> morning = "Morning"
>>> goodMorning = good + " " + morning
>>> print(good)
Good
>>> print(morning)
Morning
>>> print(goodMorning)
Good Morning
```

String Concatenation

 3_Str_Concaternation.py

File Edit Format Run Options Window Help

```
# 이름을 입력받음
name = input("-- Enter your name : ")
print("1) Hi, " + name + "님")

# 직업을 입력받음
job = input("-- Enter your job : ")
print("2) Your job is = ", job, "(String type)")

Name_Job = name + " " + job
print("3) Your name and job is = ", Name_Job)
```

```
-- Enter your name : Park
1) Hi, Park님
-- Enter your job : Engineer
2) Your job is = Engineer (String type)
3) Your name and job is = Park Engineer
^^
```


print

- '*' 연산
 - String 반복 (string repetition)
 - 숫자 곱하기 (multiplication)

```
>>>  
>>> three = "three "  
>>> print(three*3)  
three three three  
>>>  
>>> five = "five "  
>>> print(five*5)  
five five five five five  
>>>
```

print

- String 출력
 - 특수문자
 - '\n' : 새로운 라인 (new line)
 - '\t' : 탭(tab)

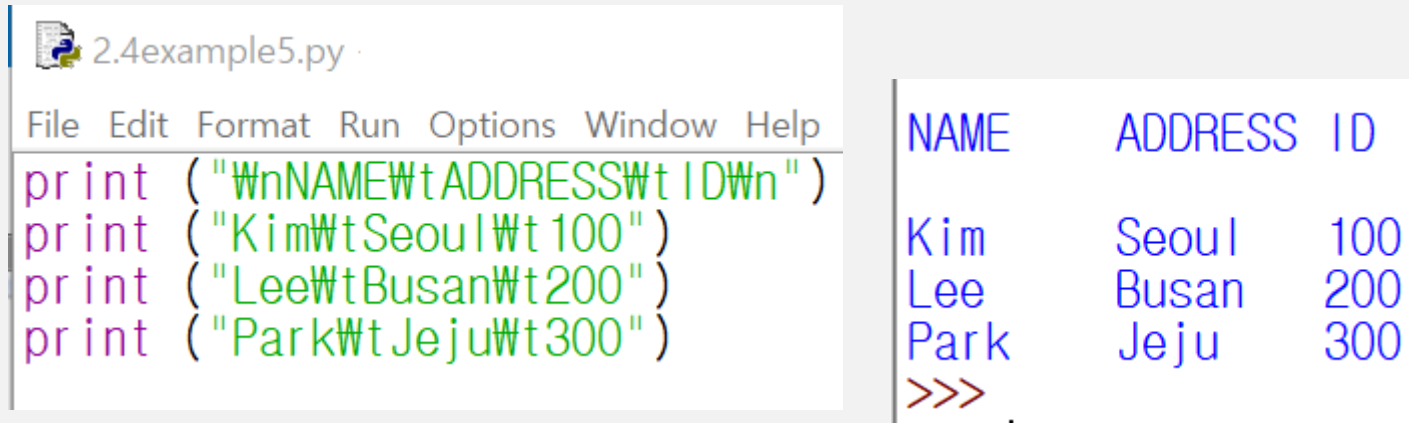
 2.4example5_1.py

```
File Edit Format Run Options Window Help
print()
print ("Good\n" + "morning !!")
print ("name address")
print ("name\taddress")
print()
```

```
Good
morning !!
name address
name      address
```

print

- String 출력
 - 특수문자
 - '\n' : 새로운 라인 (new line)
 - '\t' : 탭(tab)



The screenshot shows a Python IDE window titled '2.4example5.py'. The code in the editor is:

```
print ("\nNAME\tADDRESS\tID\n")
print ("Kim\tSeoul\t100")
print ("Lee\tBusan\t200")
print ("Park\tJeju\t300")
```

The output of the code is displayed to the right of the editor, showing a table with three columns: NAME, ADDRESS, and ID. The data rows are Kim, Seoul, 100; Lee, Busan, 200; and Park, Jeju, 300. The prompt '>>>' is visible at the bottom of the output area.

NAME	ADDRESS	ID
Kim	Seoul	100
Lee	Busan	200
Park	Jeju	300

print

- String 출력
 - Escape 문자

확장문자	내용
\'	단일 인용문자
\"	이중 인용문자
\\	역슬래시(back slash)문자
\a	Beep 소리문자
\b	백스페이스
\f	Form-feed
\n	개행문자(줄바꿈)
\r	Carriage return
\t	Tab (수평)
\v	Tab (수직)
\000	Null

print

- String 출력
 - Escape 문자
 - Ex) \'\'를 출력하고 싶을 때?
`print ("Good\\morning")`

```
print("나는 ₩ " 배가고프다₩" 라고 말했습니다.")  
print("햄버거₩피자₩치킨")  
print("₩₩ ₩₩ ₩₩ ₩₩ ₩₩")
```

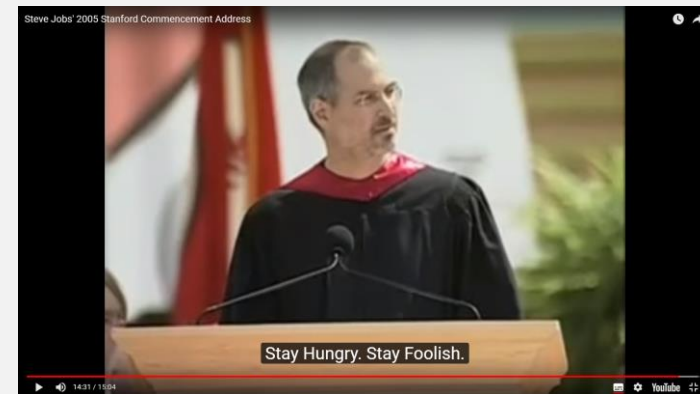
print

- 여러 줄 문자열
 - 큰 따옴표(또는 작은 따옴표)를 3번 반복해서 사용하면 여러 줄 문자열을 표현 할 수 있다.
 - \w 를 하고 줄을 바꾸면, 줄 연속을 의미한다.

```
multiline = """\w
Don't let the noise of other's opinions drown out your inner voice.
And most important, have the courage to follow your heart and intuition.
They somehow already know what you truly want to become.
Everything else is secondary.
"""

print(multiline)
```

<https://news.stanford.edu/2005/06/14/jobs-061505/>



W 출력

```
print("\n1) WW와 WWt 출력 예제")
print("\nWtGoodWWmorning\n\n")
print("\n2) WW와 WWt 출력 예제 후")
```

" " 출력

```
print("\n3) W" W" 출력 예제")
print("\nWt히딩크 감독은 W"아직도 배가고프다W"라고 말했습니다.\n")
```

multiline 출력

```
print("\n4) multiline 출력 예제")
```

multiline = """W

```
WtDonW't let the noise of otherW's opinions drown out your inner voice.
WtAnd most important, have the courage to follow your heart and intuition.
WtThey somehow already know what you truly want to become.
WtEverything else is secondary."""
```

```
print(multiline, "\n")
```

1) W와 Wt 출력 예제

GoodWmorning

2) W와 Wt 출력 예제 후

3) " " 출력 예제

히딩크 감독은 "아직도 배가고프다"라고 말했습니다.

4) multiline 출력 예제

```
Don't let the noise of other's opinions drown out your inner voice.
And most important, have the courage to follow your heart and intuition.
They somehow already know what you truly want to become.
Everything else is secondary.
```

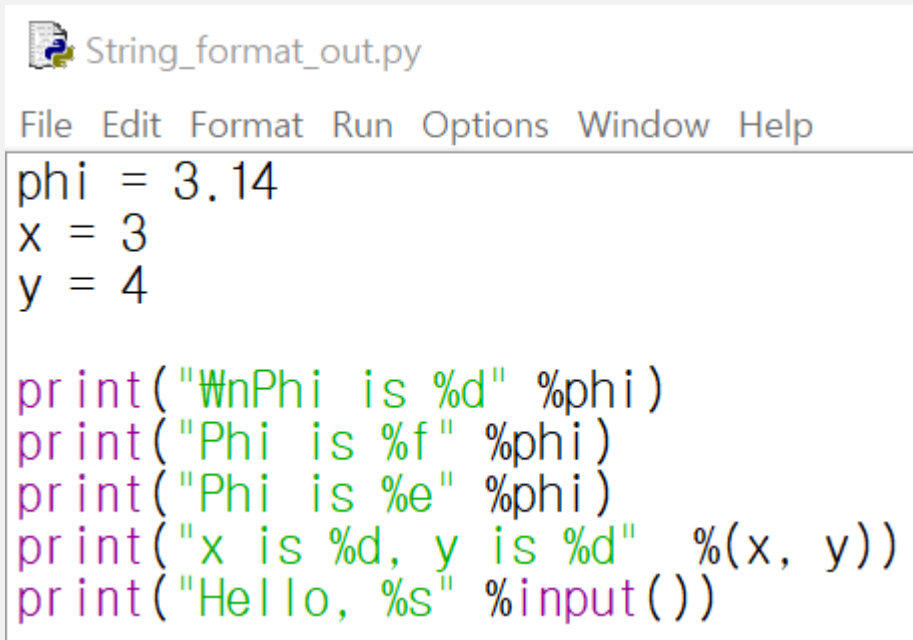
print

- String 출력 format

Format	의미
%s	String(문자열)
%c	Character(문자)
%d	Decimal(10진수 정수)
%f	Floating-point(부동소수)
%o	Octal(8진수 정수)
%x	hexa(16진수 정수. 소문자표시)
%X	hexa(16진수 정수. 대문자표시)
%e	과학적 수치(소문자표시)
%E	과학적 수치(대문자표시)

print

- String 출력 format



```
String_format_out.py
File Edit Format Run Options Window Help
phi = 3.14
x = 3
y = 4

print("\nPhi is %d" %phi)
print("Phi is %f" %phi)
print("Phi is %e" %phi)
print("x is %d, y is %d" %(x, y))
print("Hello, %s" %input())
```

```
Phi is 3
Phi is 3.140000
Phi is 3.140000e+00
x is 3, y is 4
Park
Hello, Park
>>>
```

print

- String 출력 format

```
# 문자열  
print("I have %s apples." %"five")  
  
# 문자 1개  
print("My grade is %c." %'A')  
  
# 정수  
print("I have %d apples." %3)  
  
# 부동소수 및 %  
print("Error is %f%%" %3.234)  
  
# 8진수, 16진수  
print("%o %x %X" %(9, 10, 255))
```

print

- format method
 - '{}' 형태의 기호를 포함한 문자열 뒤에 '.format()' 을 붙여서 사용

```
# 문자열
print("I have {} apples.".format("five"))

# 문자 1개
print("My grade is {}".format('A'))

# 정수
print("I have {} apples.".format(3))

# 부동소수 및 %
print("Error is {}".format(3.234))

# 8진수, 16진수
print("{} {} {}".format(9, 10, 255))
```

<https://docs.python.org/3/library/stdtypes.html?highlight=format#str.format>

```
# 문자열
print("\n1) I have %s apples." % "five")

# 문자 1개
print("2) My grade is %c." % 'A')

# 정수
print("3) I have %d apples." % 3)

# 부동소수 및 %
print("4) Error is %f" % 3.234)

# 8진수, 16진수
print("5) %o %x %X" % (9, 10, 255))

# 문자열
print("6) I have {} apples.".format("five"))

# 문자 1개
print("7) My grade is {}".format('A'))

# 정수
print("8) I have {} apples.".format(3))

# 부동소수 및 %
print("9) Error is {}".format(3.234))

# 8진수, 16진수
print("10) {} {} {}".format(9, 10, 255))
```

```
1) I have five apples.
2) My grade is A.
3) I have 3 apples.
4) Error is 3.234000
5) 11 a FF
6) I have five apples.
7) My grade is A.
8) I have 3 apples.
9) Error is 3.234
10) 9 10 255
>>>
```

print

- 선택 연산자([<숫자>])
- 범위 선택 연산자([<숫자>:<숫자>])

```
# 연결 연산자(Concatenation)  
print("안녕하세요" + "...!")
```

```
# 반복 연산자  
print("안녕하세요" * 3)
```

```
# 선택 연산자(Indexing)  
print("안녕하세요"[0])  
print("안녕하세요"[1])  
print("안녕하세요"[2])  
print("안녕하세요"[3])  
print("안녕하세요"[4])
```

```
# 범위 선택 연산자(Slicing)  
print("안녕하세요"[0:2])  
print("안녕하세요"[1:3])  
print("안녕하세요"[2:4])  
print("안녕하세요"[:3])  
print("안녕하세요"[3:])
```

print

print_string_2.py

File Edit Format Run Options Window Help

```
# 연결 연산자(Concatenation)
print("1) 안녕하세요" + "...!")
```

```
# 반복 연산자
print("2) 안녕하세요" * 3)
```

```
# 선택 연산자(Indexing)
print("안녕하세요"[0])
print("안녕하세요"[1])
print("안녕하세요"[2])
print("안녕하세요"[3])
print("안녕하세요"[4])
```

```
# 범위 선택 연산자(Slicing)
print("안녕하세요"[0:2])
print("안녕하세요"[1:3])
print("안녕하세요"[2:4])
print("안녕하세요"[:3])
print("안녕하세요"[3:])
```

```
1) 안녕하세요...!
2) 안녕하세요 2) 안녕하세요 2) 안녕하세요
안녕하세요
안녕하세요
안녕하세요
안녕하세요
안녕하세요
>>>
```

Inputting numbers

- String vs. integer vs. float
 - 문자열 age를 정수(integer)로 변환하여 저장.
 - 문자열 phi를 실수(float)로 변환하여 저장.

```
3_Str_Int_Float.py
File Edit Format Run Options Window Help
1 # 이름을 입력받음
2 somebody = input(">> 이름을 입력하세요 : ")
3 print(">> 안녕하세요, " + somebody + "님")
4
5 # 나이를 입력받아 int로 변환
6 age = int(input("\n>> 나이를 입력하세요 : "))
7 print(">> 당신의 나이 = ", age, "(Int type)")
8 print(">> 10년 후, ", somebody, "님의 나이는", age+10, "입니다")
9
10 # Phi 값을 String으로 받아 floating point로 변환
11 phi = float(input("\n>> " + somebody + "님, Phi값이 얼마지요 ? "))
12 print(">>", somebody, "님의 대답은 ", phi, "입니다")
13
14 if 3.14 <= phi < 3.15:
15     print(">>", somebody, "님, 멋지십니다 !")
16 else :
17     print(">>", somebody, "님, 다시한번 생각해 보세요.")
18
19 print("\n>> 반가왔습니다.")
20
```

```
>> 이름을 입력하세요 : 국민대
>> 안녕하세요, 국민대님
```

```
>> 나이를 입력하세요 : 22
>> 당신의 나이 = 22 (Int type)
>> 10년 후, 국민대 님의 나이는 32 입니다
```

```
>> 국민대님, Phi값이 얼마지요 ? 3.17
>> 국민대 님의 대답은 3.17 입니다
>> 국민대 님, 다시한번 생각해 보세요.
```

```
>> 반가왔습니다.
```

Inputting numbers

- 문자열을 숫자로 바꾸기
 - int 함수 또는 float 함수를 이용해서 바꾼다.
- 숫자를 문자열로 바꾸기
 - str 함수를 이용한다.

문자열 끼리 덧셈을 합니다.

```
string1 = str(123)
```

```
string2 = str(456)
```

```
print("string1 + string2:", string1 + string2)
```

정수 끼리 덧셈을 합니다.

```
number1 = int("123")
```

```
number2 = int("456")
```

```
print("number1 + number2:", number1 + number2)
```


Inputting numbers

- 실습

- 사용자로부터 두 자연수 A, B를 입력 받고, 두 자연수의 사칙연산 결과를 출력하는 코드를 작성 하세요.

```
A : 5  
B : 2  
A+B : 7  
A-B : 3  
A*B : 10  
A/B : 2.5
```

Inputting numbers

사용자로부터 두 자연수를 입력 받는다.

```
A = int(input("A : "))
```

```
B = int(input("B : "))
```

사칙 연산의 결과를 출력 합니다.

```
print("A+B :", A+B)
```

```
print("A-B :", A-B)
```

```
print("A*B :", A*B)
```

```
print("A/B :", A/B)
```

Input from the web

- World Wide Web(WWW)으로부터의 입력
 - Hyper-Text Markup Language(HTML) 파일
 - “from urllib.request import urlopen”
 - from, import: 모듈 사용 명령어
 - urllib.request, urlopen: 사용 모듈
 - url: uniform resource locator

2.4example8.py

File Edit Format Run Options Window Help

```
1 print("1) import urlopen" + "\n"*3)
2 from urllib.request import urlopen
3
4 print("2) url file open" + "\n")
5 #f = urlopen('https://www.python.org/')
6 f = urlopen('http://scrc.kookmin.ac.kr/')
7
8 print("3) print f.header" + "\n")
9 print(f.headers)
10
11 print("4) file read" + "\n")
12 html = f.read()
13
14 print("5) print html" + "\n"*2)
15 print(html)
16
17 print("\n"*2 + "6) end")
18
19
```

1) import urlopen

2) url file open

3) print f.header

Date: Wed, 20 Sep 2023 10:41:56 GMT
Set-Cookie: __wpm_client=5b90c25b31e141f684ef66c797f132ab; path=/; HttpOnly
Link: <http://scrc.kookmin.ac.kr/index.php?rest_route=/>; rel="https://api.w.org/"
Link: <http://scrc.kookmin.ac.kr/index.php?rest_route=/wp/v2/pages/572>; rel="alternate"; type="application/json"
Link: <http://scrc.kookmin.ac.kr/>; rel=shortlink
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

4) file read

5) print html

Squeezed text (1054 lines).

6) end

>>>

실습

- 난수 생성 및 출력
 - random 모듈
 - import random 사용 (모듈에 대해서는 뒤에서 설명)
 - random.randint(a,b) : a이상 b이하인 숫자 임의 생성

```
2_Random_Number.py
File Edit Format Run Options Window Help
import random

#첫 번째 random number
i = 1
print("1) Random #%d : %x" %(i, random.randint(1,1000000000000000)))

#두 번째 random number
i = i + 1
print("2) Random #%d : %x" %(i, random.randint(1,100)))

#세 번째 random number
i = i + 1
print("3) Random #%d : %x" %(i, random.randint(1,100)))

#네 번째 random number
i = i + 1
print("4) Random #%d : %x" %(i, random.randint(1,100)))

#다섯 번째 random number
i = i + 1
print("5) Random #%d : %x" %(i, random.randint(1,100)))
```

```
_Random_Number.py ===
1) Random #1 : 116693adf69
2) Random #2 : 21
3) Random #3 : 53
4) Random #4 : 60
5) Random #5 : 50
>>>
=== RESTART:
_Random_Number.py ===
1) Random #1 : 233aa20c4ca
2) Random #2 : 1c
3) Random #3 : 29
4) Random #4 : 21
5) Random #5 : 5c
>>>
=== RESTART:
_Random_Number.py ===
1) Random #1 : 289e20d467
2) Random #2 : 21
3) Random #3 : 1b
4) Random #4 : 1f
5) Random #5 : 62
>>>
```