

2주차-02

파이썬 기초 프로그래밍



표준 입출력

콘솔 입출력이란?

- 표준 입출력 장치인 **키보드**에서 데이터를 입력 받고, 표준 출력 장치인 화면으로 **데이터**를 출력하는 것
- 콘솔 입력 : `input()`
- 콘솔 출력 : `print()`

파이썬에서 표준 입력 방법들

- `sys.stdin`
 - 명령어 라인에서 직접 입력 받음
- `input()`
 - 사용자로 부터 입력 받음
- `fileinput.input()`
 - 파일로 부터 입력 받음

sys.stdin

```
import sys

for line in sys.stdin:
    if 'q' == line.rstrip():
        break
    print(f'Input : {line}')

print("Exit")
```

```
ayushman@ayushman-Lenovo-G50-80:~/Documents/gfg/stdin$ python3 stdin_test.py
Geeksforgeeks
Input : Geeksforgeeks

No
Input : No

Okay
Input : Okay

q
Exit
```

input

```
inp = input("Type anything")  
print(inp)
```

```
ayushman@ayushman-Lenovo-G50-80:~/Documents/gfg/stdin$ python3 input.py  
Type Anything: GeeksForGeeks :)  
GeeksForGeeks :)
```

input()

사용법

- `s = input()`
 - 커서가 깜박이며 입력을 기다림
- `name = input('이름을 입력하세요 : ')`
 - '이름을 입력하세요 : '를 출력한 후 커서가 깜박이며 입력을 기다림

```
>>> s = input()
윤소정
>>> name = input('이름을 입력하세요 : ')
이름을 입력하세요 : 윤소정
>>> s
'윤소정'
>>> name
```

입력값

- 문자열
 - 123을 입력하면 문자열 '123' 이 저장됨
- `i = (input("정수를 입력하세요 : "))`
 - 입력한 값을 정수로 저장할 때는
정수 변환 함수를 사용해야 함!!

```
>>> i = input('정수를 입력하세요 : ')
정수를 입력하세요 : 123
>>> i
'123'
>>> i = int(input('정수를 입력하세요 : '))
정수를 입력하세요 : 123
>>> i
123
>>>
```

fileinput

```
import fileinput
with fileinput.input(files = ('sample.txt', 'no.txt')) as f:
    for line in f:
        print(line)
```

```
ayushman@ayushman-Lenovo-G50-80:~/Documents/gfg/stdin$ python3 stdin_test.py
Geeksforgeeks
Input : Geeksforgeeks

No
Input : No

Okay
Input : Okay

q
Exit
```


print()

- `print('Hello World')` # 문자열 출력
- `print(a)` # 변수 값 출력
- `print('계산 결과는', result, '입니다')` # 문자열과 변수 값 출력
- `print(100+100)` # 연산 결과값 출력
- `print('값 : %d' % sum)` # 서식 출력

print() : 인수

- end : 마지막 출력 문자 지정 (기본값은 \n)
 - `print(1,2, end=' ')`
 - `print(3,4)`
- sep : 항목 간의 출력 문자 지정 (기본값은 공백)
 - `print(1,2,3,4,5)`
 - `print(1,2,3,4,5, sep=',')`
- file : 파일 객체로 출력 (기본값은 표준출력)
 - `f = open('out.txt', 'w')`
 - `print(1,2,3,4,5, file=f)`
 - `f.close()`
 - `open('out.txt').read()`

print() – 서식 출력(1)

- 사용법

- print(서식문자열 %(서식대응값))

서식	값의 예	설명
%d, %x, %o	10, 100, 1234	정수(10진수, 16진수, 8진수)
%f	0.5 , 1.0 , 3.14	실수(소수점이 붙은 수)
%c	“b”, “한”	문자 한 글자
%s	“안녕”, “abcdefg”, “a”	한 글자 이상의 문자열

- 사용예

- print('이름 : %s, 나이 : %d' %(name, age))

print() – 서식 출력(2)

- 실습

- data = 123
- print('입력하신 값은 %5d 입니다' % data)
- print('%05d' % data)
- print('%d, %x, %o' % (data, data, data))

- data = 123.45
- print('%f' % data)
- print('%10.1f' % data)
- print('%-10.3f' % data)



```
>>> data = 123.45
>>> print('%f : 값 출력' % data)
>>> print('%10.1f : 값출력' % data)
>>> print('%-10.3f : 값 출력' % data)
```

- data = 'Python'
- print('%s' % data)
- print('%10s' % data)

- ▶ name = '윤소정'
- ▶ age = 20
- ▶ print('%-10s:%10d' % (name, age))

print() – string() 메소드

```
cstr = "I love geeksforgeeks"
```

```
# 중앙 정렬 출력
```

```
# string with fillchr
```

```
print("Center aligned string with fillchr: ")
```

```
print(cstr.center(40, '#'))
```

```
# 왼쪽 정렬 출력
```

```
# string with "-" padding
```

```
print("The left aligned string is : ")
```

```
print(cstr.ljust(40, '-'))
```

```
#오른쪽 정렬 출력
```

```
# with "-" padding
```

```
print("The right aligned string is : ")
```

```
print(cstr.rjust(40, '-'))
```

다양한 특수 문자(1)

- 이스케이프(escape) 문자들
 - \ (역슬래시)로 시작
 - `print("줄 바꾸기 \n 줄 바꾸기")`

이스케이프 문자	역할	비고
\n	새로운 줄로 이동	 키 누른 효과
\t	다음 탭으로 이동	 키 누른 효과
\\	\ 출력	
\'	' 출력	
\"	" 출력	

다양한 특수 문자(2)

- 사용 예

- `print('\t 탭키 \t 연습')`
- `print("I'm a student")`
- `print('작은 따옴표 안에서 \' 사용하기\!')`
- `print('\\\\\\\\ 역슬래시 3개 출력')`
- `print(r'\n \t \' \\ 를 그대로 출력')`

- 실행 후 결과 확인!!!

- 다시 복습...

```
파이썬의 세계에 오신 것을 환영합니다.  
파이썬은 참 쉽고 강력합니다.  
나는 파이썬이 좋습니다.  
>>>
```

format()

using format() method

```
print('I love {} for "{}!"'.format('Geeks', 'Geeks'))
```

```
print('{0} and {1}'.format('Geeks', 'Portal'))
```

```
print('{1} and {0}'.format('Geeks', 'Portal'))
```


format()

combining positional and keyword arguments

```
print("number one portal is {0}, {1}, and {other}.".format("Geeks", "For", other="Geeks"))
```

using format() method with number

```
print("Geeks :{0:2d}, Portal :{1:8.2f}".format(12, 00.546))
```

Changing positional argument

```
print("Second argument: {1:3d}, first one: {0:7.2f}".format(47.2, 11))
```

```
print("Geeks: {a:5d}, Portal: {p:8.2f}".format(a = 453, p = 59.058))
```