
Python

Function

함수/사용자 입력과 출력

파일읽고쓰기

함수

함수/사용자 입력과 출력

```
In [2]: 1 def sum(a,b):  
        2     return a+b
```

```
In [3]: 1 a = 3  
        2 b = 4  
        3 c = sum(a,b)  
        4 print(c)
```

7

```
In [4]: 1 # 입력값이 없는 함수  
        2 def say():  
        3     return 'Hi'
```

```
In [5]: 1 a = say()  
        2 print(a)
```

Hi

함수

함수/사용자 입력과 출력

```
In [6]: 1 # 결과값이 없는 함수
        2 def sum(a,b):
        3     print("%d, %d의 합은 %d입니다." % (a,b,a+b))
```

```
In [7]: 1 sum(3,4)

3, 4의 합은 7입니다.
```

```
In [8]: 1 a = sum(3,4)

3, 4의 합은 7입니다.
```

```
In [9]: 1 print(a)

None
```

```
In [10]: 1 # 입력값도 결과값도 없는 함수
        2 def say():
        3     print('Hi')
```

```
In [11]: 1 say()

Hi
```

```
In [12]: 1 # 여러 개의 입력값을 받는 함수 만들기
          2 def sum_many(*args):
          3     sum = 0
          4     for i in args:
          5         sum = sum + i
          6     return sum
```

```
In [13]: 1 result = sum_many(1,2,3)
          2 print(result)
```

6

```
In [15]: 1 result = sum_many(1,2,3,4,5,6,7,8,9,10)
          2 print(result)
```

55

함수

```
In [16]: 1 # 입력인수가 두개
          2 def sum_mul(choice, *args):
          3     if choice == "sum":
          4         result = 0
          5         for i in args:
          6             result = result + i
          7     elif choice == "mul":
          8         result = 1
          9         for i in args:
          10             result = result * i
          11     return result
```

```
In [17]: 1 result = sum_mul('sum', 1,2,3,4,5)
          2 print(result)
```

15

```
In [18]: 1 result = sum_mul('mul', 1,2,3,4,5)
          2 print(result)
```

120

함수

함수/사용자 입력과 출력

```
In [19]: 1 # 함수의 결과값(리턴값)
          2 def sum_and_mul(a,b):
          3     return a+b, a*b
```

```
In [21]: 1 result = sum_and_mul(3,4)
          2 print(result) # 튜플값(a+b, a*b) 리턴
          (7, 12)
```

```
In [26]: 1 sum, mul = sum_and_mul(3,4)
          2 print("%d, %d" % (sum, mul))
          7, 12
```

```
In [29]: 1 def say_nick(nick):
          2     if nick == "바보":
          3         return
          4     print("나의 별명은 %s입니다." % nick)
```

```
In [30]: 1 say_nick('야호')
          나의 별명은 야호입니다.
```

```
In [31]: 1 say_nick('바보')
```

```
In [32]: 1 # 입력인수에 초깃값 미리 설정하기
2 def say_myself(name, old, man=True):
3     print("나의 이름은 %s입니다" % name)
4     print("나이는 %d입니다." % old)
5     if man:
6         print("남자입니다.")
7     else:
8         print("여자입니다.")
```

```
In [33]: 1 say_myself("박항서", 27)
```

나의 이름은 박항서입니다
나이는 27입니다.
남자입니다.

```
In [35]: 1 say_myself("박항서", 27, True)
```

나의 이름은 박항서입니다
나이는 27입니다.
남자입니다.

```
In [36]: 1 say_myself("박항서", 27, False)
```

나의 이름은 박항서입니다
나이는 27입니다.
여자입니다.


```
In [37]: 1 a = 1
          2 def vartest(a):
          3     a = a + 1
          4
          5 vartest(a)
          6 print(a)
```

1

```
In [38]: 1 a = 1
          2 def vartest(a):
          3     a = a + 1
          4     return a
          5
          6 vartest(a)
          7 print(a)
```

1

```
In [40]: 1 a = 1
          2 def vartest():
          3     global a # global 명령어 사용하지 않는 것이 좋다.
          4     a = a + 1
          5
          6 vartest()
          7 print(a)
```

2

사용자 입력과 출력

함수/사용자 입력과 출력

```
In [1]: # 입력되는 모든 것은 문자열로 취급  
a = input()
```

Life is too short, you need python

```
In [5]: a
```

```
Out [5]: 'Life is too short, you need python'
```

```
In [3]: number = input("숫자를 입력하세요.")
```

숫자를 입력하세요.3

```
In [6]: print(number)
```

3

```
In [1]: # 큰 따옴표(")로 둘러싸인 문자열은 +연산과 동일  
print("life" "is" "too short")  
lifeistoo short
```

```
In [2]: print("life" + "is" + "too short")  
lifeistoo short
```

```
In [3]: # 문자열 띄어쓰기는 콤마로 한다  
print("life", "is", "too short")  
life is too short
```

```
In [4]: # 한 줄에 결과값 출력하기  
for i in range(10):  
    print(i, end=' ')  
0123456789
```

함수/사용자 입력과 출력

파일읽고쓰기

파일 읽고 쓰기

```
In [1]: # 파일 생성
f = open("새파일.txt", 'w')
f.close()
```

```
In [3]: # 파일을 쓰기 모드로 열어 출력값 적기
f = open("C:/Temp/새파일.txt", 'w')
for i in range(1,11):
    data = "%d번째 줄입니다.\n" % i
    f.write(data)
f.close()
```

```
In [4]: for i in range(1,11):
        data = "%d번째 줄입니다.\n" % i
        print(data)
```

1번째 줄입니다.

2번째 줄입니다.

3번째 줄입니다.

4번째 줄입니다.

5번째 줄입니다.

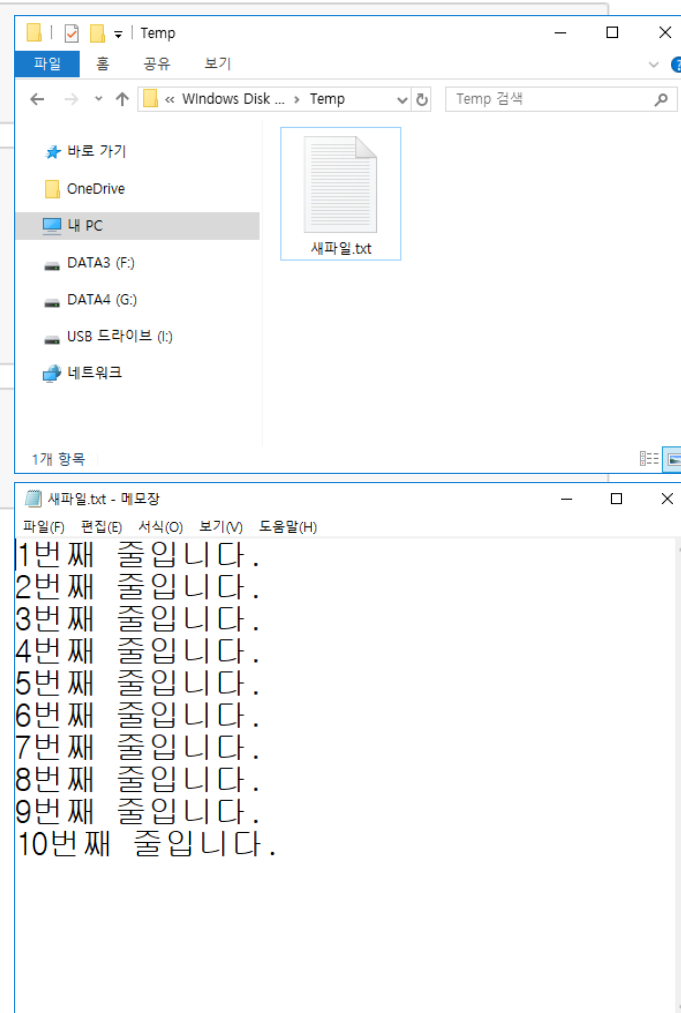
6번째 줄입니다.

7번째 줄입니다.

8번째 줄입니다.

9번째 줄입니다.

10번째 줄입니다.



파일 읽기 : readline() 함수

```
In [5]: f = open("C:/Temp/새파일.txt", 'r')
        line = f.readline()
        print(line)
        f.close()
```

1번째 줄입니다.

```
In [6]: f = open("C:/Temp/새파일.txt", 'r')
        while True:
            line = f.readline()
            if not line: break
            print(line)
        f.close()
```

1번째 줄입니다.

2번째 줄입니다.

3번째 줄입니다.

4번째 줄입니다.

5번째 줄입니다.

6번째 줄입니다.

7번째 줄입니다.

8번째 줄입니다.

9번째 줄입니다.

10번째 줄입니다.

파일읽기:readlines()함수

```
In [7]: # readlines() : 파일의 모든 라인을 읽어서 각각의 줄을 요소로 갖는 리스트로 리턴
f = open("C:/Temp/새파일.txt", 'r')
lines = f.readlines()
for line in lines:
    print(line)
f.close()
```

1번째 줄입니다.

2번째 줄입니다.

3번째 줄입니다.

4번째 줄입니다.

5번째 줄입니다.

6번째 줄입니다.

7번째 줄입니다.

8번째 줄입니다.

9번째 줄입니다.

10번째 줄입니다.

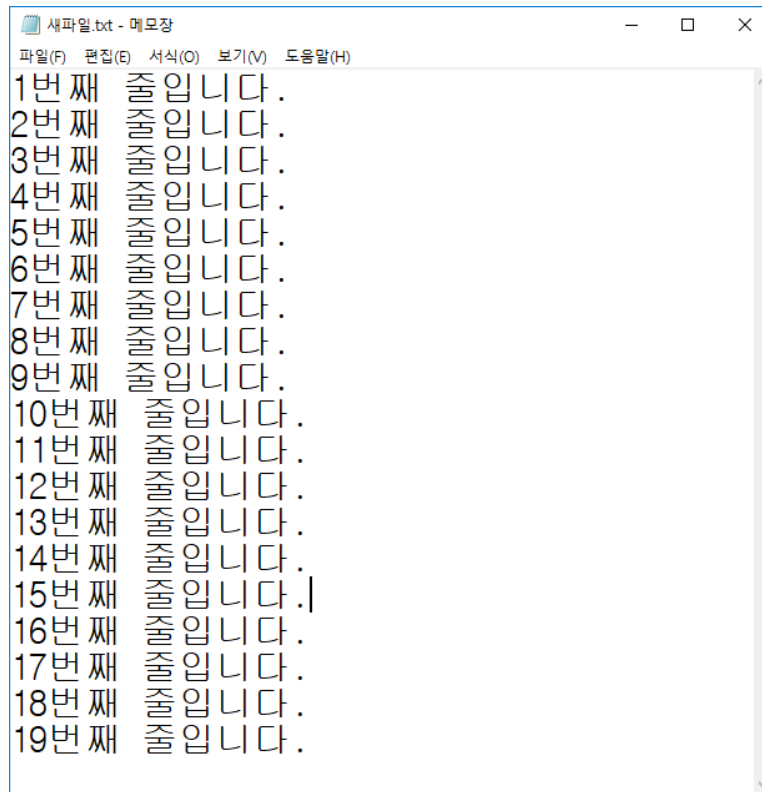
파일읽기 : read()함수

```
In [8]: # read() : 파일의 내용 전체를 문자열로 리턴  
f = open("C:/Temp/새파일.txt", 'r')  
data = f.read()  
print(data)  
f.close()
```

1번째 줄입니다.
2번째 줄입니다.
3번째 줄입니다.
4번째 줄입니다.
5번째 줄입니다.
6번째 줄입니다.
7번째 줄입니다.
8번째 줄입니다.
9번째 줄입니다.
10번째 줄입니다.

파일에 내용 추가

```
In [9]: f = open("C:/Temp/새파일.txt", 'a')
        for i in range(11,20):
            data = "%d번째 줄입니다.\n" % i
            f.write(data)
        f.close()
```



with문

파일 읽고 쓰기

```
In [11]: f = open("foo.txt", 'w')
          f.write("Life is too short, you need python")
          f.close()
```

```
In [12]: # with문은 파일을 열고 닫는 것을 close()필요없이 자동으로 처리해 준다.
          with open("foo.txt", "w") as f:
              f.write("Life is too short, you need python")
```

sys 모듈로 입력 인수 주기

[파일 읽고 쓰기](#)

```
In [13]: # 입력인수를 직접 줄 수 있는 방법
# sys1.py
import sys
args = sys.argv[1:]
for i in args:
    print(i)

# 실행예제
# C:\python>python sys1.py aaa bbb ccc
# aaa
# bbb
# ccc
```