



파일 입출력

Update – 2018.06

Contents

- 사용자 입력과 출력
- 파일 읽고 쓰기
- `readline()`, `readlines()`
- `with` 문 이용하기

사용자 입력과 출력

input의 사용

```
Input('Message')
```

프롬프트를 띄워서 사용자 입력 받기

```
>>> number = input("숫자를 입력하세요: ")
숫자를 입력하세요: 3
>>> print(number)
3
>>>
```

사용자 입력과 출력

print 문

```
>>> print("life" "is" "too short")
lifeistoo short
>>> print("life"+"is"+"too short")
lifeistoo short
```

```
>>> print("life", "is", "too short")
life is too short
```

```
>>> for i in range(10):
...     print(i, end=' ')
...
0 1 2 3 4 5 6 7 8 9
```

파일 읽고 쓰기

파이썬의 File I/O

```
f = open( " 파일경로 " , '파일접근모드' , encoding='uft-8')  
f.write('데이터')  
f.close()
```

파일 접근 모드	설명
r	읽기모드 - 파일을 읽기만 할 때 사용
w	쓰기모드 - 파일에 내용을 쓸 때 사용
a	추가모드 - 파일의 마지막에 새로운 내용을 추가 시킬 때 사용

파일 읽고 쓰기

파일 생성하기

```
f = open("data/test.txt", 'w')  
f.close()
```

주의사항 -
대상 폴더는 미리
생성되어 있어야 한다.

파일을 쓰기 모드로 열어 출력 값 적기

```
f = open("data/test.txt", 'w')  
for i in range(1, 11):  
    data = "%d번째 줄입니다.\n" % i  
    f.write(data)  
f.close()
```

파일 읽고 쓰기

파일에 새로운 내용 추가하기

```
f = open('data/test.txt', 'a')
f.write('-'*50)
for i in range(11,21):
    data = '\n %d : Hello Python ' % i
    f.write(data)
f.write('\n'+ '-'*50)
f.close()
```

파일 읽고 쓰기

read() 함수 이용하기

```
f = open("data/test.txt", 'r')
data = f.read()
print(data)
print(type(data))
f.close()
```

String 객체
<class 'str'>

readline() / readlines()

readline() 함수

f.readline()은 파일의 첫번째 줄을 읽어 출력한다.

```
f = open("data/yesterday.txt", 'r')
line = f.readline()
print(line)
print(type(line))
f.close()
```

String 객체
<class 'str'>

readline() / readlines()

```
myfile = open('data/Yesterday.txt','r')
while True:
    line = myfile.readline()
    if not line:
        break
    print(line)
f.close()
```

String 객체
<class 'str'>

모든 라인을 읽어서 화면에
출력한다.

readline() / readlines()

readlines() 함수 이용하기

```
f = open("C:/Python/새파일.txt", 'r')
lines = f.readlines()
for line in lines:
    print(line)
f.close()
```

#List객체
<class 'str'>

모든 라인을 읽어서 화면에
출력한다.

readline() / readlines()

모든 라인을 읽어서 화면에 출력한다.

```
myfile = open('data/Yesterday.txt','r')
textList = myfile.readlines()
print(textList)
print(type(textList))
print('-'*50,'\n')
for i in textList:
    print(i)
myfile.close()
print('-'*50,'\n')
```

with 문 사용하기

with문과 함께 사용하기

: 파일 Open() 후 close() 명령줄을 삽입하지 않아도 된다.

```
with open('파일명', '파일 액세스 모드') as 파일변수:  
    .... 명령문 ...
```

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

```
with open("foo.txt", "w") as f:  
    f.write("Life is too short, you need python")
```

