

Python 07

김은진



List comprehension

- To derive a new list from an existing list, doing iterative operations on the list.

[expr **for** var **in** list *filtering*]

```
[x/2.0 for x in [4,5,6]]  
print( [a[0]+'!' for a in [['Hal','Abelson'], ['Jacob','White'],  
                           ['Leslie', 'Kaelbling']]])
```

```
listone = [2,3,4]  
listtwo = [2*i for i in listone if i > 2]  
print(listtwo)
```

```
nums = [1,2,5,6,88,99,101,10000,100,37,101]  
print([x for x in nums if x%2==1])
```

```
>>> =====  
>>>  
['Hal!', 'Jacob!', 'Leslie!']  
[6, 8]  
[1, 5, 99, 101, 37, 101]
```



Nested list comprehension

```
>>> x_pts = [-1, 0, 2]
>>> y_pts = [2, 4]
>>> xy_pts = [[x, y] for x in x_pts for y in y_pts]
>>> print(xy_pts)
[[-1, 2], [-1, 4], [0, 2], [0, 4], [2, 2], [2, 4]]
```



File과 directory(폴더)

이름	수정한 날짜	유형	크기
TextInput	2018-03-07 오후...	파일 폴더	
tracing	2018-01-10 오후...	파일 폴더	
twain_32	2018-01-10 오후...	파일 폴더	
Vss	2018-01-10 오후...	파일 폴더	
Web	2018-01-10 오후...	파일 폴더	
WinSxS	2018-04-18 오후...	파일 폴더	
AhnInst.log	2018-01-15 오후...	텍스트 문서	122KB
bfsvc.exe	2017-09-29 오후...	응용 프로그램	64KB
bootstat.dat	2018-04-29 오후...	DAT 파일	66KB
Dtcinstall.log	2018-01-10 오후...	텍스트 문서	2KB
explorer.exe	2018-02-10 오후...	응용 프로그램	3,813KB

Directory

File



File 사용하기

1. open 함수 호출, 반환값 변수에 저장
2. 쓰기, 읽기 함수 호출
3. close 함수 호출

```
>>> =====
>>>
Programming is fun
When the work is done
if you wanna make your work also fun:
use Python!
```

poem.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Programming is fun
When the work is done
if you wanna make your work also fun:
use Python!

Zero length indicates EOF

```
poem = '''\
Programming is fun
When the work is done
if you wanna make your work also fun:
use Python!
'''
```

```
# Open for 'w'riting
1 f = open('poem.txt', 'w')
# Write test to file
3 f.write(poem)
# Close the file
4 f.close()
```

```
# If no mode is specified,
# 'r' mode is assumed
1 f = open('poem.txt')
while True:
2 # Read one line
3 line = f.readline()
# Zero length indicates EOF
4 if len(line) == 0:
    break
# The line already has a newline
# at the end of each line
# since it is reading from a file.
print(line, end='')
#close the file
f.close()
```



File 열기 / 닫기

```
f = open('fun.txt', 'w')
```

File이름

작업모드

File열기: built-in 함수 호출

```
f.close()
```

File닫기: 작업대상 함수 호출

작업 모드	설명
r	읽기모드 – 파일을 읽기만 할 때 사용
w	쓰기모드 – 파일에 데이터를 쓸 때 사용
a	추가모드 – 파일의 마지막부터 새로운 데이터를 추가 할 때 사용
r+	읽기 + 쓰기 모드



File에 쓰기

```
f = open('fun.txt', 'w')
```

```
f.write('Programming is fun.')
```

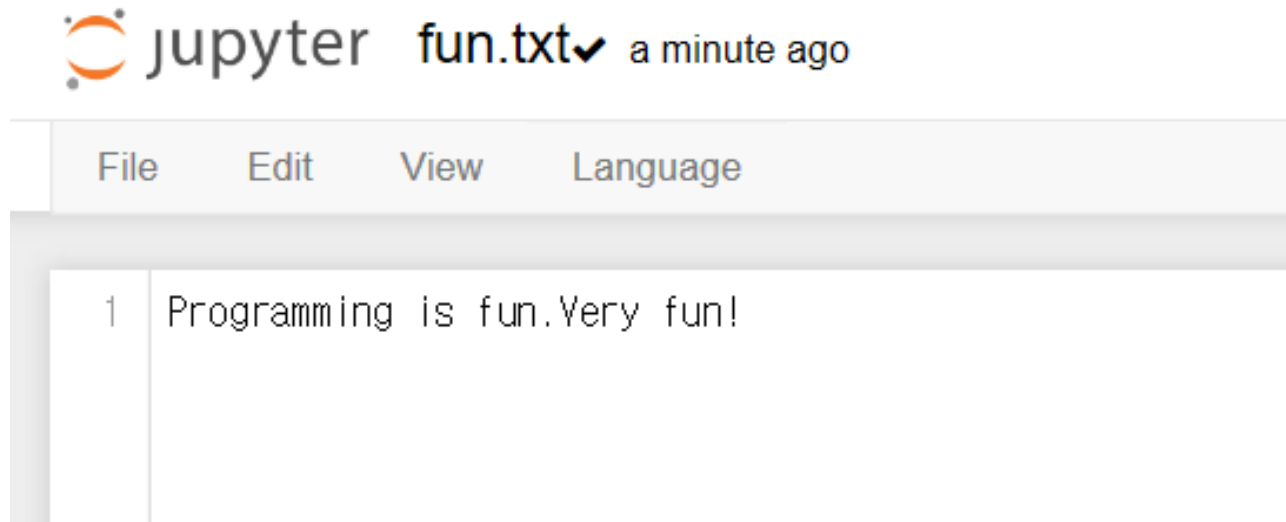
wirte(): 작업대상 함수 호출

```
f.write('Very fun!')
```

```
f.close()
```


동일한 프로그램 2회 실행해 보기!!

기존 file에 추가 쓰기 안되고 덮어쓰기



File에 쓰기|2

```
f = open('fun.txt', 'w' )  
f.write('Programming is fun.\n')  
f.write('Very fun!')  
f.close()
```

 jupyter fun.txt ✓ a few seconds ago

File Edit View Language

```
1 Programming is fun.  
2 Very fun!
```



File 한 줄 읽기

```
f = open("fun.txt", 'r')
```

```
line = f.readline()
```


readline(): 작업대상 함수 호출

```
print(line)
```

한 줄 읽기

```
f.close()
```

마지막에 있는 enter 까지 str에 넣음

 jupyter fun.txt ✓ a few seconds ago

File Edit View Language

```
1 Programming is fun.  
2 Very fun!
```



File 전체 읽기 - 반복문

```
f = open("fun.txt", 'r')
while True :
    line = f.readline()
    if not line:
        break
    print(line)
f.close()
```

더 이상 읽은 데이터가
없는 경우 종료



File 전체 한번에 읽기

```
f = open("fun.txt", 'r')
```

```
lines = f.readlines()
```

```
for line in lines:
```

```
    print(line)
```

```
f.close()
```

모든 라인을 읽어서
각 라인이 하나의 원소가 되는
list를 리턴



File 전체 한번에 읽기2

```
f = open("fun.txt", 'r')
```

```
lines = f.read()
```

```
print(lines)
```

```
f.close()
```

모든 라인을 읽어서
내용 전체를 하나의 문자열로 리턴



File에 추가로 쓰기

```
f = open("fun.txt", 'a' )  
f.write('very very fun ')  
f.close()
```

Append모드 (추가 쓰기)

jupyter fun.txt ✓ a few seconds ago

File	Edit	View	Language
1	Programming is fun.		
2	Very fun!		

File	Edit	View	Language
1	Programming is fun.		
2	Very fun!very very fun		



close 함수 없이 함수 닫기

사용할 변수 명

```
with open("fun.txt", 'a') as f :  
    f.write('very very fun ')
```

with를 빠져 나올 때 스스로 close() 호출



os.path module의 File 관련 함수

```
import os
```

```
fullpath = r'C:\Windows\notepad.exe'
```

```
print(os.path.exists(fullpath))
```

File 존재 확인하기

```
print(os.path.dirname(fullpath))
```

directory 경로만 추출하기 (C:\Windows)

```
print(os.path.basename(fullpath))
```

파일 이름만 추출하기 (notepad.exe)

```
print(os.path.split(fullpath))
```

디렉토리 와 파일로 구분 ('C:\Windows', 'notepad.exe')

```
print(os.path.splitext(fullpath))
```

확장자와 나머지로 구분 ('C:\Windows\notepad', '.exe')



CSV(Comma-Separated Values)

- Csv file 의 예

연안정지 관측자료 다운로드 <https://goo.gl/qNho5q>

해역	<input type="text" value="전체"/>	연안정점	<input type="text" value="전체"/>
관측일시	<input type="text" value="20180514"/> ~ <input type="text" value="20180521"/>		
<div>1주일 1개월 6개월</div>			

20180514 ~ 20180521

① 입력

Q 검색	↓ 텍스트	↓ 엑셀	↓ CSV
------	-------	------	-------

② 선택

③ 선택

총 건수 : 40 건

해역	연안정점	관측일시 (YYYYMMDD)	수온 [℃]	수온 QC flag	기온 [℃]	기온 QC	운량	천기
동해	감포(12001)	20180514		2	19.9	2	11%-20%	쾌청(운량:0-2)



File 위치

- 현재 작업 directory 에 file 옮기기
import os
a = os.getcwd() 작업 directory
- File의 fullname 주기
 - File의 속성 창에서 위치 정보가 directory명
 - r"director명\file명" 을 file이름으로 사용



csv file

- CSV(Comma-separated values)
 - 각 라인의 컬럼들의 구분 문자가 콤마(,)로 분리된 텍스트 File 포맷
- 메모장에 열어보면

```
"해역","연안정점","관측일시","수온","수온 QC Flag","기온","기온 QC","운량","천기"
"동해","감포(12001)","20170501","","2","16.5","2","11%-20%","쾌청(운량:0-2)"
"동해","감포(12001)","20170502","","2","15.2","2","31%-40%","개임(운량:3-7)"
"동해","감포(12001)","20170503","","2","16","2","21%-30%","개임(운량:3-7)"
"동해","감포(12001)","20170504","","2","19.5","2","21%-30%","개임(운량:3-7)"
"동해","감포(12001)","20170505","","2","20.2","2","91%-100%","흐림(운량:8-10)"
"동해","감포(12001)","20170506","","2","20.3","2","11%-20%","쾌청(운량:0-2)"
"동해","감포(12001)","20170507","","2","19.5","2","21%-30%","개임(운량:3-7)"
"동해","감포(12001)","20170508","","2","24.3","2","21%-30%","개임(운량:3-7)"
```

헤더

데이터행



csv 화일읽기

```
import csv
```

```
filename='연안정지 관측자료.csv'
```

```
with open(filename, 'r', encoding='utf-8') as f:
```

```
    reader = csv.reader(f) 한글이 포함된 경우
```

reader 만들기

```
    row = next(reader)
```

행 1개 읽기

```
    print(row)
```



for 문으로 한 줄 씩 읽기

```
import csv
filename='연안정지 관측자료.csv'

with open(filename, 'r', encoding='utf-8') as f :
    reader = csv.reader(f)
    for row in reader:
        print(row)
```



헤더는 skip, 데이터 행들만 출력

```
# -*- coding: utf-8 -*-
```

```
import csv
```

```
filename='연안정지 관측자료.csv'
```

```
with open(filename,  
            newline="", encoding="utf-8" ) as f:
```

```
    r = csv.reader(f)
```

```
    next(r) 첫번째 헤더는 skip
```

```
    for row in r: 데이터 행만 출력
```

```
        print(row)
```



csv file에 쓰기

```
import csv
```

```
with open('out.csv', 'w', newline="",  
        encoding="utf-8" ) as f:
```

newline="" 필수

(Windows인 경우에
라인 뒤 빈 문자가 추가되는 문
제)

```
writer = csv.writer(f, delimiter=',')  
writer 만들기
```

```
writer.writerow([1, 2, 3, 4, 5])
```

행 1개 쓰기



CSV이므로, 구분 문자(delimiter) = ','

```
"해역","연안정점","관측일시","수온","수온 QC Flag","기온","기온 QC","운량","천기"
"동해","감포(12001)","20170501","","2","16.5","2","11%-20%","쾌청(운량:0-2)"
"동해","감포(12001)","20170502","","2","15.2","2","31%-40%","개임(운량:3-7)"
"동해","감포(12001)","20170503","","2","16","2","21%-30%","개임(운량:3-7)"
"동해","감포(12001)","20170504","","2","19.5","2","21%-30%","개임(운량:3-7)"
"동해","감포(12001)","20170505","","2","20.2","2","91%-100%","흐림(운량:8-10)"
"동해","감포(12001)","20170506","","2","20.3","2","11%-20%","쾌청(운량:0-2)"
"동해","감포(12001)","20170507","","2","19.5","2","21%-30%","개임(운량:3-7)"
"동해","감포(12001)","20170508","","2","24.3","2","21%-30%","개임(운량:3-7)"
```



list 연산 이용하여 csv file 쓰기

```
import csv
with open('eggs.csv', 'w', newline='') as f:
    w = csv.writer(f, delimiter=',')
    w.writerow(['Spam'] * 5 + ['Baked Beans'])
    w.writerow(['Spam', 'Lovely Spam', 'Wonderful Spam'])
```

eggs.csv file

```
Spam,Spam,Spam,Spam,Spam,Baked Beans
Spam,Lovely Spam,Wonderful Spam
```



csv file 쓰기 : 행, 열 확인

```
import csv
```

```
with open('eggs.csv', 'w', newline='') as f:
```

```
    w = csv.writer(f, delimiter=',')
```

```
    w.writerow(['Spam'] * 5 + ['Baked Beans'])
```

```
    w.writerow(['Spam', 'Lovely Spam', 'Wonderful Spam'])
```

행=list

열=list의 item



tsv file 쓰기

- TSV(Tab-separated values)
 - 각 라인의 column들의 구분 문자가 Tab(\t)으로 분리된 텍스트 File 포맷

```
import csv
```

```
with open('eggs.tsv', 'w', newline='') as f:
```

```
    w = csv.writer(f, delimiter='\t' )
```

```
    w.writerow(['Spam'] * 5 + ['Baked Beans'])
```

```
    w.writerow(['Spam', 'Lovely Spam', 'Wonderful Spam'])
```

eggs.tsv

Spam	Spam	Spam	Spam	Spam	Baked Beans
Spam	Lovely	Spam	Wonderful	Spam	



tsv file 읽기

```
import csv
with open("eggs.tsv", 'r', newline="") as f:
    r = csv.reader(f, delimiter='\t')
    for row in r:
        print(row)
```

```
['Spam', 'Spam', 'Spam', 'Spam', 'Spam', 'Baked Beans']
['Spam', 'Lovely Spam', 'Wonderful Spam']
```



File

- File
 - File 열기 / 닫기 : `open()`, `close()`, `with`
 - File 읽기 : `readline()`, `readlines()`, `read()`
 - File 쓰기 : `write()`
- CSV File
 - CSV 읽기 / 쓰기
- TSV File
 - TSV 읽기 / 쓰기

