

Python Workshop

- Raspberry Pi -

2017년 8월
서강대학교

Python File IO

Python을 파일 읽고 쓰기

- <https://wikidocs.net/26>
 - Writing
 - `f = open("내파일.txt", 'w')`
 - `f.write(data)`
 - Reading
 - `f = open("내파일.txt", 'r')`
 - `data = f.read()`
 - `line = f.readline()`
 - `lines = f.readlines()`
 - Append
 - `f = open(" 내파일.txt",'a')`
 - `f.write(data)`

Python

-List, Tuple, Dictionary, Set

Container Types

- Tuple <https://wikidocs.net/15>
 - 예) (4, 1, 0, 8)
 - Ordered, Constant, indexing, slicing, +, *
- List <https://wikidocs.net/14>
 - 예) [5, 3, 9, 1]
 - Ordered, indexing, slicing, +, *, modification, deletion, append, sort, reverse sort, index, insert, remove, pop, count, extend
- Dict <https://wikidocs.net/16>
 - 예) {1:"kim", 4:"Lee"}
 - Key-Value, Unique Key, Unordered, append, deletion, Key검사
- Set <https://wikidocs.net/1015>
 - 예) {2, 4, 1, 6}
 - Unique Value, Unordered, Union, Intersection, Difference, addition, update, remove

List, Tuple, Dictionary, Set

- <https://wikidocs.net/14> : 리스트
- <https://wikidocs.net/15> : 튜플
- <https://wikidocs.net/16> : 딕셔너리
- <https://wikidocs.net/1015> : 집합

String Methods

```
str.capitalize()
str.casefold()
str.center(width[, fillchar])
str.count(sub[, start[, end]])
str.encode(encoding="utf-8", errors="strict")
str.endswith(suffix[, start[, end]])
str.expandtabs(tabsize=8)
str.find(sub[, start[, end]])
str.format(*args, **kwargs)
str.format_map(mapping)
str.index(sub[, start[, end]])
str.isalnum()
str.isalpha()
str.isdecimal()
str.isdigit()
str.isidentifier()
str.islower()
str.isnumeric()
str.isprintable()
str.isspace()
str.istitle()
str.isupper()
```

```
str.join(iterable)
str.ljust(width[, fillchar])
str.lower()
str.lstrip([chars])
str.partition(sep)
str.replace(old, new[, count])
str.rfind(sub[, start[, end]])
str.rindex(sub[, start[, end]])
str.rjust(width[, fillchar])
str.rpartition(sep)
str.rsplit(sep=None, maxsplit=-1)
str.rstrip([chars])
str.split(sep=None, maxsplit=-1)
str.splitlines([keepends])
str.startswith(prefix[, start[, end]])
str.strip([chars])
str.swapcase()
str.title()
str.translate(table)
str.upper()
str.zfill(width)
```

Dict Methods

```
len(d)  
d[key]  
d[key] = value  
del d[key]  
key in d  
key not in d  
iter(d)  
clear()  
copy()  
classmethod fromkeys(seq[, value])  
get(key[, default])  
items()  
keys()  
pop(key[, default])  
popitem()  
setdefault(key[, default])  
update([other])  
values()
```


Tuple Type -- <https://wikidocs.net/71>

```
a = 1, 5, 23, 7
print(type(a))
print(a)
|
```

```
<class 'tuple'>
(1, 5, 23, 7)
```

- 튜플은 값을 변경할 수 없다 → Constant!
 - 그러나, 튜플과 리스트 간의 변환은 자유롭다

```
a = 1, 5, 23, 7
print(type(a))
print(a)
#del a[2]

b = list(a)
print(type(b))
del b[2]
print(b)

c = tuple(b)
print(type(c))
print(c)
```

Regular Expression - <https://wikidocs.net/4308>

Regular Expression

- 특정 패턴을 추출

```
import re
p = re.compile('Wd{8}')

f=open("x.txt", "r")
d = f.readlines()
f.close()

for k in d:
    k = k.replace('\n', '')
    m = p.search(k)
    if (m):
        print(m.group(1))
    else:
        print(k)

f.close()
```

```
Abklekfh49874433alkdjhfiieer
Dkfle8rldfj98464387oeirfjlaskdf
a
Laskdf lasdkf;a3457823sldkdkfja
;s
owdjf23987465halksjdfnas;df
As,kdfhias;odjfhalslskdf287649
87
```

Regular Expression

- 패턴 정의
 - . -- matches any character
 - ^ -- matches the start of the string
 - \$ -- matches the end of string
 - * -- 0 or more repetitions
 - + -- 1 or more repetitions
 - ? -- 0 or 1 repetitions
 - {m} – m copies
 - [] – individual listing [ake] a or k or e
 - \d – matches a decimal digit
 - \w – matches an alpha-numeric ([a-zA-Z0-9])
 - \s – matches a space character
- Online Regular Expression Tester
 - <https://regex101.com/>

Str type에 대한 연산

- Indexing
 - `a[2]`
- Slicing
 - `a[3, 7]`, `a[:3]`, `a[3:]`, `a[:]`, `a[-1]`
- Concatenation (plus)
 - `"kim" + "kyuho"`
- Copy (multiplication)
 - `'-' * 60`
- ~~Replace~~ → create new string
 - `b = mystring.replace("you", "me")`

Advanced Topics of "str"

- Basic methods
 - <https://docs.python.org/3.5/library/stdtypes.html#textseq>

```
n = 1
for j in range(4,0,-1):
    for i in range(j):
        print(n, j, i)
    n += 1
```

```
a = "kim Kyuho"
b = a.replace("kim", "lee");
print(a, b)

print(a.find("kim"))

b = "{} {} {}".format("kim kyuho", "eats", "apple")
print(b)
```

List Type - <https://wikidocs.net/70>

- Creation
 - a = [2, 5, 4, 6]
 - b = []
- Assign
 - a[2] = 9
- Append
 - a.append(5)
- Delete
 - del a[2]
- Sort
 - a.sort()
 - a.sort(reverse=True)
- List in a list
 - a[2] = [4,12,6]

```
a = [2, 5, 4, 6]
b = []
print(a)
print(b)

a[2] = 9
print(a)

a.append(8)
print(a)

del a[0]
print(a)

a.sort()
print(a)

a.sort(reverse=True)
print(a)

a[2] = [1,4,6,5]
print(a)

a[0] = ("Kim", 45)
print(a)

a[3] = "Kyuho"
print(a)

#a.sort()
#print(a)
```

Str과 List

```
a = "kim kyuho"  
b = list(a)  
print(b)  
|
```

```
['k', 'i', 'm', ' ', 'k', 'y', 'u', 'h', 'o']
```

```
c = str(b)  
print(type(c))  
print(c)  
|
```

```
<class 'str'>  
['k', 'i', 'm', ' ', 'k', 'y', 'u', 'h', 'o']
```


Programming Projects

1. 파일에서 숫자를 읽어들인다
 - 숫자의 개수, 합을 계산한다.
 - Sequence가 빠진 것들을 찾아내어 출력한다.(1,2,3,5,6) 이라면 4가 빠진것이다.
2. 파일에서 한 라인씩 읽어들인다.
 - 각 라인에서 숫자와 숫자가 아닌 글자를 구분하여 2 칼럼으로 출력한다.
3. 숫자 맞추기 게임을 한다.
 - 답이 현재 숫자보다 높으면 하얀색LED를, 아래쪽이면 LED를 끄고, 컴퓨터가 이기면 파란색 LED를, 사람 플레이어가 이기면 빨간색 LED를 켜다.
4. Your Idea?

Program #1: file read

프로그램 실행전에 미리 동일한 폴더에 'mylog.txt'라는 빈 파일을 만들어 둡니다.

```
f = open('mylog.txt', 'r+')
```

```
original = f.read()
```

```
print("저장된 메모")
```

```
print(original)
```

```
name = input("이름: ")
```

```
comment = input("코멘트: ")
```

```
data = name + ": " + comment + '\n'
```

```
f.write(data)
```

```
f.close()
```

```
print('새 메모 등록완료 >> ', data)
```

Program #2: fetch web page saving as a file

```
import urllib.request
```

```
f = urllib.request.urlopen('http://finance.naver.com/item/frgn.nhn?code=005930')
```

```
data = f.read().decode('euc-kr')
```

```
print(len(data))
```

```
f = open("naver.com", "w")
```

```
f.write(data)
```

```
f.close()
```

Program #3: fetch web page and extract a part

```
import datetime, time
import urllib.request

f = urllib.request.urlopen('http://finance.naver.com/item/frgn.nhn?code=005930')
data = f.read().decode('euc-kr')

i = data.find("<dd>현재가 ")
if i > 0:
    line = data[i:i+100]
    line = line.split(" ")
    f = open("삼성주식추적.txt", "a")
    print(datetime.datetime.now(), "삼성전자주식", line[1], file=f)
    print(datetime.datetime.now(), "삼성전자주식", line[1])
    f.close()
else:
    print(r.status, "got", len(data), "types")
```

Program #4: 자료분석

- <http://www.work.go.kr/jobMain.do> 유망직업 연봉표
- <https://1drv.ms/w/s!AtwGHPDGDtFaqiHgh-0Mm4BMmm6H> 연봉표

생명정보학자 [임금 (5,611만원), 전망 (90%)]
무인경비원 [임금 (2,907만원), 전망 (87.9%)]
다이어트프로그래머 [임금 (2,793만원), 전망 (83.3%)]
프로게이머 [임금 (5,664만원), 전망 (73.3%)]
요리강사 [임금 (3,116만원), 전망 (71%)]
임상연구코디네이터 [임금 (3,179만원), 전망 (70%)]
전직지원전문가 [임금 (3,202만원), 전망 (65.6%)]
온실가스인증심사원 [임금 (4,399만원), 전망 (63.3%)]
관광통역안내원 [임금 (3,777만원), 전망 (63.3%)]

...

- 관심주제
 - 임금과 전망의 GAP이 제일 큰 직업들을 찾아내보자.
 - 이러한 직업은 무슨 특이성이 있는 것인가?

```
f = open("salary.txt", "r")
d = f.readlines()
f.close()

i = 0
for s in d:
    i = i + 1
    #print(i, salary)
print("total ", i, "records")

f1 = open("highsalary.txt", "w")
f2 = open("brightfuture.txt", "w")
```

```
for s in d:
    i = i + 1
    data = s.split()
    job = data[0]
    salary = data[2]
    future = data[4]

    salary = salary.replace('(', '')
    salary = salary.replace(')', '')
    salary = salary.replace('만원', '')
    salary = salary.replace(',', '')
    future = future.replace('(', '')
    future = future.replace(')', '')
    future = future.replace(']', '')
    future = future.replace('%', '')
    #print(job, salary, future)
    if int(salary) > 6000:
        print("고소득", job, salary, future, file=f1)
    else:
        if float(future) > 80.0:
            print("유망성", job, salary, future, file=f2)

f1.close()
f2.close()
```