

AI Programming

Lecture 14

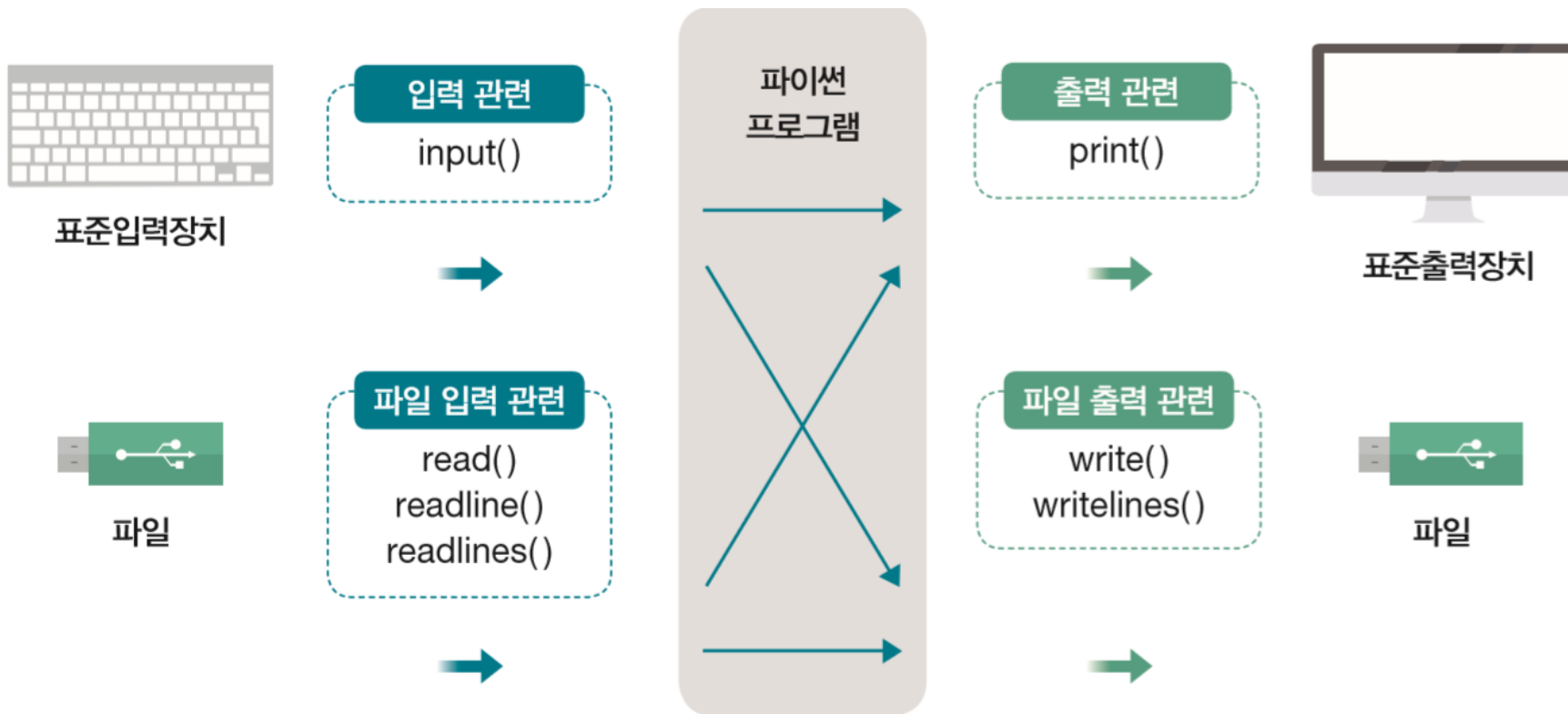
Preview

- Ch. 11 파일 입출력
 - 11.2 파일 입출력의 기본
 - 11.3 텍스트 파일 입출력

11.2 파일 입출력의 기본

Basics

- Console: keyboard + screen



Basics

- Basics of file I/O



```
f = open(filename, mode)
```

```
f.close()
```

Basics

- `open(filename, mode)`

읽기용 : 변수명 = open("파일명", "r")

쓰기용 : 변수명 = open("파일명", "w")

종류	설명
생략	r과 동일하다.
r	읽기 모드, 기본값이다.
w	쓰기 모드, 기존에 파일이 있으면 덮어쓴다.
r+	읽기/쓰기 겸용 모드이다.
a	쓰기 모드, 기존에 파일이 있으면 이어서 쓴다. append의 약어이다.
t	텍스트 모드, 텍스트 파일을 처리한다. 기본값이다.
b	이진 모드, 이진 파일을 처리한다.

11.3 텍스트 파일 입출력


Text File I/O

- `readline()`

Code11-01.py

```
1  inFp = None          # 입력 파일
2  inStr = ""           # 읽어 온 문자열
3
4  inFp = open("C:/Temp/data1.txt", "r")
5
6  inStr = inFp.readline()
7  print(inStr, end = "")
```

```
9  inStr = inFp.readline()
10 print(inStr, end = "")
11
12 inStr = inFp.readline()
13 print(inStr, end = "")
14
15 inFp.close()
```

 data1.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

CookBook 파이썬을 공부합니다.

완전 재미있어요. ^^

파이썬을 공부하기 잘했네요~~|

출력 결과

CookBook 파이썬을 공부합니다.

완전 재미있어요. ^^

파이썬을 공부하기 잘했네요~~

Text File I/O

- `readline()`

```
1  inFp = None          # 입력 파일
2  inStr = ""           # 읽어 온 문자열
3
4  inFp = open("C:/Temp/data1.txt", "r")
5
6  while True :
7      inStr = inFp.readline()
8      if inStr == "" :
9          break;
10     print(inStr, end = "")
11
12  inFp.close()
```

출력 결과

CookBook 파이썬을 공부합니다.
완전 재미있어요. ^^
파이썬을 공부하기 잘했네요~~

Text File I/O

- `with~as`
 - Do not need to close the file

```
with open("C:/Temp/data1.txt", "r") as inFp :  
    inList = inFp.readlines()  
    print(inList)
```

Text File I/O

- `readlines()`

Code11-03.py

```
1  inFp = None
2  inList = ""
3
4  inFp = open("C:/Temp/data1.txt", "r")
5
6  inList = inFp.readlines()
7  print(inList)
8
9  inFp.close()
```

출력 결과

```
['CookBook 파이썬을 공부합니다.\n', '완전 재미있어요. ^^ \n', '파이썬을 공부하기 잘했네요~~']
```

Text File I/O

- `readlines()`

Code11-04.py

```
1  inFp = None
2  inList, inStr = [], ""
3
4  inFp = open("C:/Temp/data1.txt", "r")
5
6  inList = inFp.readlines()
7  for inStr in inList :
8      print(inStr, end = "")
9
10 inFp.close()
```

출력 결과

CookBook 파이썬을 공부합니다.
완전 재미있어요. ^^
파이썬을 공부하기 잘했네요~

Text File I/O

- Directory

Code11-05.py

```
1  inFp = None
2  fName, inList, inStr = "", [], ""
3
4  fName = input("파일명을 입력하세요 : ")
5  inFp = open(fName, "r")
6
7  inList = inFp.readlines()
8  for inStr in inList :
9      print(inStr, end = "")
10
11 inFp.close()
```

출력 결과

파일명을 입력하세요 : C:/Windows/win.ini
; for 16-bit app support
[fonts]
... 생략 ...

Text File I/O

- read()

```
inFp = None  
fName, inList, inStr = "", [ ], ""
```

```
fName = input("파일명을 입력하세요 : ")
```

```
inFp = open(fName, "r")
```

```
inStr = inFp.read()  
print(inStr, end = "")
```

```
inFp.close()
```

```
파일명을 입력하세요 : C:/Temp/data1.txt  
CookBook 파이썬을 공부합니다.  
완전 재미있어요. ^^  
파이썬을 공부하기 잘했네요~~
```

Text File I/O

- Handling FileNotFoundError

출력 결과

파일명을 입력하세요 : **C:/abc.txt** ← 없는 파일

Traceback (most recent call last):

File "C:\CookPython\Code11-05.py", line 5, in <module>

inFp = open(fName, "r")

FileNotFoundError: [Errno 2] No such file or directory: 'C:/abc.txt'

Text File I/O

- Handling FileNotFoundError

Code11-06.py

```
1 import os
2
3 inFp = None
4 fName, inList, inStr = "", [], ""
5
6 fName = input("파일명을 입력하세요 : ")
7
8 if os.path.exists(fName) :
9     inFp = open(fName, "r")
10
11     inList = inFp.readlines()
12     for inStr in inList :
13         print(inStr, end = "")
14
15     inFp.close()
```

```
16 else :
17     print("%s 파일이 없습니다." % fName)
```

출력 결과

파일명을 입력하세요 : C:/abc.txt ← 없는 파일
C:/abc.txt 파일이 없습니다.

Text File I/O

- `writelines()`

Code11-07.py

```
1 outFp = None
2 outStr = ""
3
4 outFp = open("C:/Temp/data2.txt", "w")
5
6 while True :
7     outStr = input("내용 입력 : ")
8     if outStr != "" :
9         outFp.writelines(outStr + "\n")
10    else :
11        break
12
13 outFp.close()
14 print("--- 정상적으로 파일에 씀 ---")
```

출력 결과

내용 입력 : 파이썬을

내용 입력 : 열공하고

내용 입력 : 있습니다. ^^

내용 입력 :

--- 정상적으로 파일에 씀 ---

Text File I/O

- Copy

Code11-08.py

```
1  inFp, outFp = None, None
2  inStr = ""
3
4  inFp = open("C:/Windows/win.ini", "r")
5  outFp = open("C:/Temp/data3.txt", "w")
6
7  inList = inFp.readlines()
8  for inStr in inList :
9      outFp.writelines(inStr)
10
11  inFp.close()
12  outFp.close()
13  print("--- 파일이 정상적으로 복사되었음 ---")
```

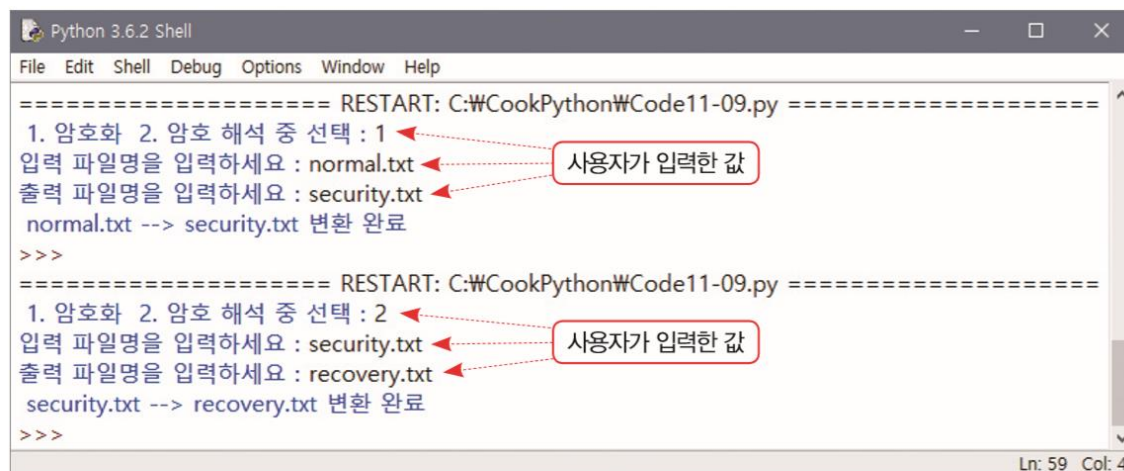
출력 결과

--- 파일이 정상적으로 복사되었음 ---

Text File I/O

- Exercise

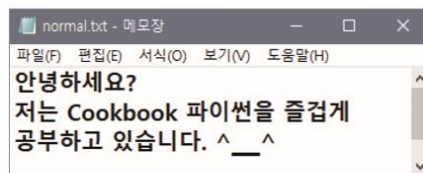
- 파일 암호화 및 암호 해독



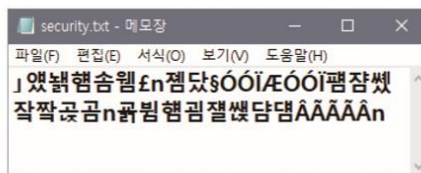
```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help

===== RESTART: C:\CookPython\Code11-09.py =====
1. 암호화 2. 암호 해석 중 선택 : 1
입력 파일명을 입력하세요 : normal.txt
출력 파일명을 입력하세요 : security.txt
normal.txt --> security.txt 변환 완료
>>>

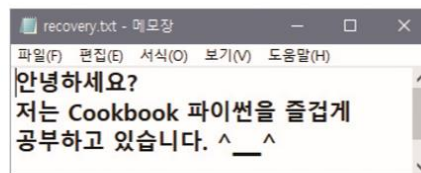
===== RESTART: C:\CookPython\Code11-09.py =====
1. 암호화 2. 암호 해석 중 선택 : 2
입력 파일명을 입력하세요 : security.txt
출력 파일명을 입력하세요 : recovery.txt
security.txt --> recovery.txt 변환 완료
>>>
```



```
normal.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
안녕하세요?
저는 Cookbook 파이썬을 즐겁게
공부하고 있습니다. ^__^
```



```
security.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
J 앓뵡햐몹햐몹꺆n햐몹땡꺆ÓÓĬÆÓÓĬ햐몹햐몹
작작꺆몹꺆n꺆햐몹햐몹햐몹햐몹햐몹햐몹햐몹햐몹
```



```
recovery.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
안녕하세요?
저는 Cookbook 파이썬을 즐겁게
공부하고 있습니다. ^__^
```

원본 파일 → 암호화된 파일 → 다시 암호 해독된 파일

Text File I/O

- ASCII code

- `ord('A')` = 65

- `ord('a')` = 97

10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자
64	0x40	@	80	0x50	P	96	0x60	~	112	0x70	p
65	0x41	A	81	0x51	Q	97	0x61	a	113	0x71	q
66	0x42	B	82	0x52	R	98	0x62	b	114	0x72	r
67	0x43	C	83	0x53	S	99	0x63	c	115	0x73	s
68	0x44	D	84	0x54	T	100	0x64	d	116	0x74	t
69	0x45	E	85	0x55	U	101	0x65	e	117	0x75	u
70	0x46	F	86	0x56	V	102	0x66	f	118	0x76	v
71	0x47	G	87	0x57	W	103	0x67	g	119	0x77	w
72	0x48	H	88	0x58	X	104	0x68	h	120	0x78	x
73	0x49	I	89	0x59	Y	105	0x69	i	121	0x79	y
74	0x4A	J	90	0x5A	Z	106	0x6A	j	122	0x7A	z
75	0x4B	K	91	0x5B	[107	0x6B	k	123	0x7B	{
76	0x4C	L	92	0x5C	\	108	0x6C	l	124	0x7C	
77	0x4D	M	93	0x5D]	109	0x6D	m	125	0x7D	}
78	0x4E	N	94	0x5E	^	110	0x6E	n	126	0x7E	~
79	0x4F	O	95	0x5F	_	111	0x6F	o	127	0x7F	DEL

Text File I/O

- **Exercise**

- `ord()` : return ASCII code of input string
- `chr()` : return string of input ASCII code

```
ord('파')  
chr(54028)
```

출력 결과

```
54028  
'파'
```

Text File I/O

- **Exercise**

- Encoding: ASCII + security key
 - Example: 54028('파') + 100 = 54123('팸')
- Decoding: ASCII - security key

```
num = ord('팸')  
chr(num - 100)
```

출력 결과

'파'

Text File I/O

Code11-09.py

```
1  ## 변수 선언 부분 ##
2  inFp, outFp = None, None
3  inStr, outStr = "", ""
4  i = 0
5  secu = 0
6
7  ## 메인 코드 부분 ##
8  secuYN = input(" 1. 암호화 2. 암호 해석 중 선택 : ")
9  inFname = input("입력 파일명을 입력하세요 : ")
10 outFname = input("출력 파일명을 입력하세요 : ")
11
12 if secuYN == "1" :
13     secu = 100
14 elif secuYN == "2" :
15     secu = -100
16
17 inFp = open(inFname, 'r', encoding = 'utf-8')
```

```
18 outFp = open(outFname, 'w', encoding = 'utf-8')
19
20 while True :
21     inStr = inFp.readline()
22     if not inStr :
23         break
24
25     outStr = ""
26     for i in range(0, len(inStr)) :
27         ch = inStr[i]
28         chNum = ord(ch)
29         chNum = chNum + secu
30         ch2 = chr(chNum)
31         outStr = outStr + ch2
32
33     outFp.write(outStr)
34
35 outFp.close()
36 inFp.close()
37 print('%s --> %s 변환 완료' % (inFname, outFname))
```

Summary

- **File I/O**

- `f=open(filename, mode)~f.close()`
- `readline()`, `readlines()`, `read()`
- `writelines()`

Assignment 12

• 텍스트 파일 word counter

- countCharacter(inFp)
 - 파일 객체 inFp의 단어 수 count
 - Dictionary 변수 freq 반환
- writeCountFile(freq, outFp)
 - Dictionary 변수 freq를 파일 객체 outFp에 write
 - 더 많이 등장한 순서대로 write
- findMaxKey(dict)
 - Dictionary 변수 dict의 가장 큰 value에 해당하는 key 반환

```
## 함수 코드 부분 ##
def countCharacter(inFp):
    freq = {}

    ## 코드 작성 ##

    return freq # dictionary return

def writeCountFile(freq, outFp):
    ## 코드 작성 ##

def findMaxKey(dict):
    maxKey, maxValue = 0, 0
    for k in dict.keys():
        if dict[k] > maxValue:
            maxKey = k
            maxValue = dict[k]
    return maxKey

def main():
    inFp = open("mobydick.txt", "r")
    freq = countCharacter(inFp)
    inFp.close()

    print(freq)

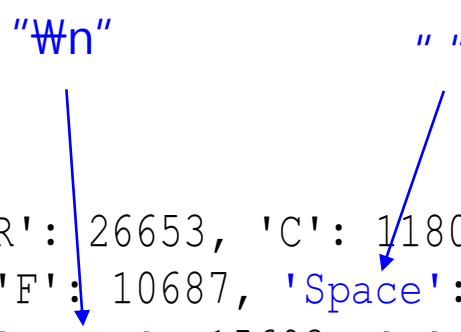
    outFp = open("count.txt", "w")
    writeCountFile(freq, outFp)
    outFp.close()

## 메인 코드 부분 ##
main()
```

Assignment 12

- (Cont'd)

- Dictionary 출력 예시




```
{ 'I': 34034, 'P': 8962, 'U': 13730, 'R': 26653, 'C': 11806, 'B': 8485,
  'A': 40032, 'S': 32531, 'E': 59646, 'F': 10687, 'Space': 115314,
  'O': 35898, 'T': 44879, 'L': 21690, 'Enter': 15603, 'N': 33754, 'V': 4512,
  '?': 464, '{': 2, 'D': 19279, '&': 3, 'Q': 915, 'W': 11322, ',': 9536,
  '.': 4246, '1': 410, '8': 99, '9': 81, '2': 222, 'H': 31750, 'K': 4195,
  'M': 12336, 'Y': 8810, '-': 2290, 'G': 10708, '3': 162, '6': 96, '4': 162,
  'X': 665, '0': 116, '5': 116, '7': 89, ';': 1958, '!': 739, '"': 2431,
  '(': 136, ')': 133, '*': 127, 'J': 596, ':': 99, 'Z': 270, '^': 17, '}' : 3,
  '<': 5, '$': 3, "'": 323, '/': 55, '~': 2, '\\': 15, '_': 2, '>': 5, '|': 1 }
```

Assignment 12

- (Cont'd)

- count.txt 파일 예시

 count.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

|space --> 115314

E --> 59646

T --> 44879

A --> 40032

O --> 35898

I --> 34034

N --> 33754

S --> 32531

H --> 31750

R --> 26653

L --> 21690

D --> 19279

Enter --> 15603