

(12주차 실습)

11장(파일입출력) 실습

2019. 11. 19.

오늘의 실습 (11장 실습)

- 다음을 Python으로 코딩하고, 소스코드와 실행결과의 capture화면을 제출하라.
 - 11장
 - 338페이지 SELF STUDY 11-1
 - 339페이지 SELF STUDY 11-2
 - 343페이지 SELF STUDY 11-3
 - 344페이지 SELF STUDY 11-4
 - 368~370 페이지 응용예제 01 파일탐색기 만들기
- 필요한 입력 데이터파일은 만들어 사용하라.
- 제출 : 11. 25.(월) 자정까지
- 제출물 : 1개의 아래아한글 또는 pdf 보고서로 제출
 - 소스코드는 실행해 볼 수 있도록 표 안에 붙여넣기로 제출
 - 실행결과는 소스코드와 실행결과를 볼 수 있게 화면을 capture하여 제출

338페이지 SELF STUDY 11-1

- Code11-02.py를 수정해서 다음과 같이 앞에 각 행번호를 출력해 보자.

Code11-02.py

```
inFp = None      # 입력 파일
inStr = ""       # 읽어온 문자열
```

```
inFp = open("C:/Temp/data1.txt", "r")
```

```
while True :
    inStr = inFp.readline()
    if inStr == "" :
        break;
    print(inStr, end = "")
```

```
inFp.close()
```

```
al\Programs\Python\Python37\1.py ==
1 : CookBook 파이썬을 공부합니다.
2 : 완전 재미있어요.^.^
3 : 파이썬을 공부하기 잘했네요~~
>>> |
```

339페이지 SELF STUDY 11-2

- Code11-04.py를 수정해서 다음과 같이 앞에 각 행의 번호를 출력해 보자.

Code11-04.py

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

```
inFp = open("C:/Temp/data1.txt", "r")
```

```
inList = inFp.readlines()  
for inStr in inList :  
    print(inStr, end="")
```

```
inFp.close()
```

```
al\Programs\Python\Python37\1.py ===  
1 : CookBook 파이썬을 공부합니다.  
2 : 완전 재미있어요.^.  
3 : 파이썬을 공부하기 잘했네요~~  
>>> |
```

343페이지 SELF STUDY 11-3

- Code11-07.py를 수정해서 파일명도 입력받아보자.
 - 힌트 : Code11-05.py를 참고한다

Code11-07.py

```
outFp = None
outStr = ""

outFp = open("C:/Temp/data2.txt", "w")

while True:
    outStr = input("내용 입력 : ")
    if outStr != "" :
        outFp.writelines(outStr + "\n")
    else :
        break

outFp.close()
print("--- 파일에 정상적으로 써졌음 ---")
```

Code11-05.py

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

fName = input("파일명을 입력하세요 : ")
inFp = open(fName, "r")

inList = inFp.readlines()
for inStr in inList :
    print(inStr, end = "")

inFp.close()
```

344페이지 SELF STUDY 11-4

- Code11-08.py를 수정해서 두 파일 모두 파일명을 입력받아 보자

Code11-08.py

```
inFp, outFp = None, None
inStr = ""

inFp = open("C:/Windows/win.ini", "r")
outFp = open("C:/Temp/data3.txt", "w")

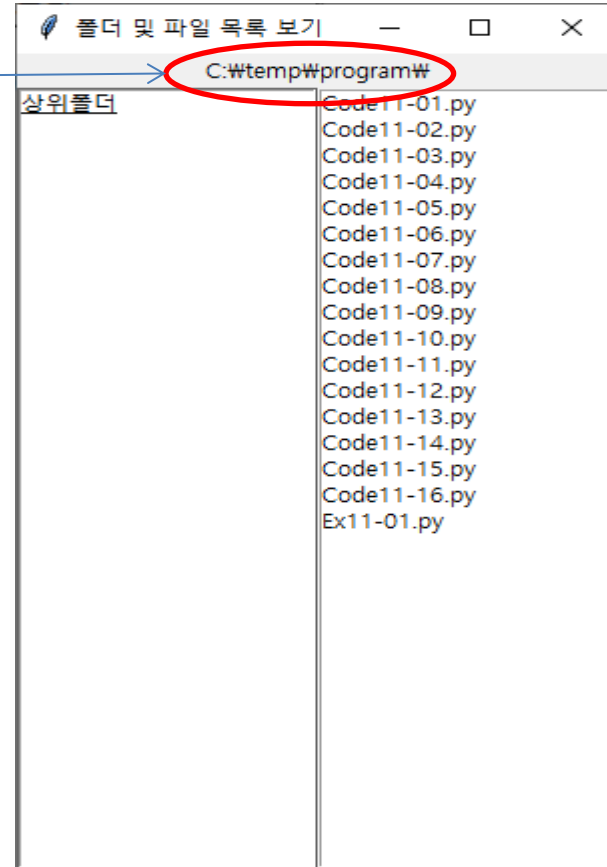
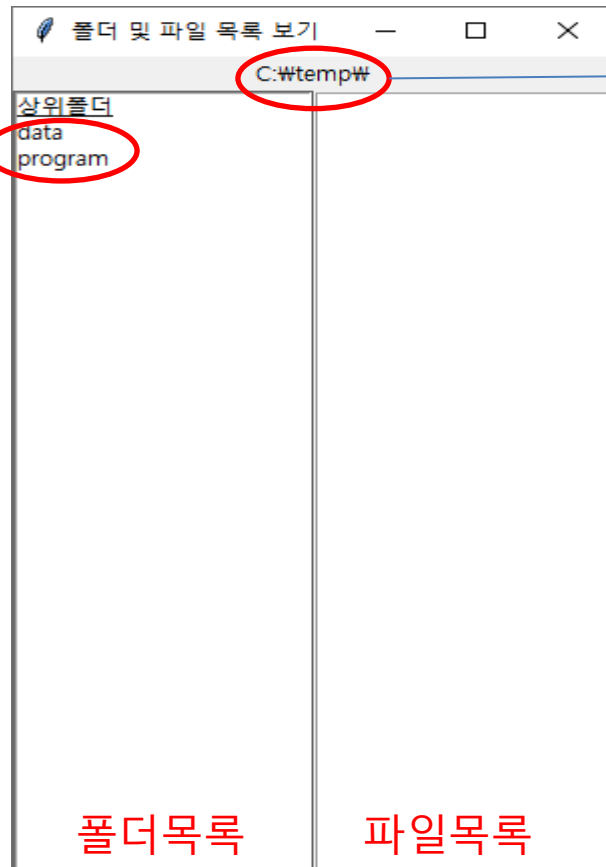
inList = inFp.readlines()
for inStr in inList :
    outFp.writelines(inStr)

inFp.close()
outFp.close()
print("--- 파일이 정상적으로 복사되었음 ---")
```

```
입력받을 화일명을 입력하시오 : c:/windows/win.ini
출력할 화일명을 입력하시오 : c:/temp/data4.txt
--- c:/windows/win.ini 파일이 c:/temp/data4.txt 로 복사되었음 ---
```

368~370 페이지 응용예제 01 파일 탐색기 만들기 (1/8)

- 기준 폴더를 C:WW에서 **C:WWtempWW**로 바꾸어보라



368~370 페이지 응용예제 01 파일 탐색기 만들기 (2/8)

- 윈도의 파일탐색기와 기능이 비슷한 프로그램을 만들어 보자. 그림을 보면
 - 위쪽에는 현재 폴더명을 표시하고,
 - 왼쪽에는 현재 폴더의 하위폴더 목록을 표시하며,
 - 오른쪽에는 현재 폴더의 파일목록을 표시한다.
 - 또, 왼쪽의 폴더를 클릭하면 해당 폴더로 이동하고,
 - '상위폴더'를 클릭하면 상위폴더로 이동한다.

368~370 페이지 응용예제 01 파일 탐색기 만들기 (3/8)

```
1      from tkinter import *
2      import os
3      import os.path
4
5      ## 함수 선언 부분 ##
6      def clickListBox(evt):
7          global currentDir, searchDirList
8          if (dirListBox.curselection() == () ) : # 다른 리스트 박스를 클릭할 때는 무시함.
9              return
10         dirName =str(dirListBox.get(dirListBox.curselection())) # 클릭한 폴더 이름
11         if dirName == '상위폴더' :
12             if len(searchDirList) == 1 : # 상위 폴더를 클릭했는데, 현재 C:₩₩면 무시함.
13                 return
14             searchDirList.pop() # 상위 폴더 이동이므로, 마지막 검색 폴더(=현재 폴더) 제거
15         else :
16             searchDirList.append(currentDir+dirName+'₩₩') # 검색 리스트에 클릭한 폴더 추가
17
18         fillListBox()
19
20     def fillListBox() : # 항상 제일 검색한 폴더 리스트의 마지막 폴더(=현재 폴더)를 표시
21         global currentDir, searchDirList, dirLabel, dirListBox, fileListBox
22         dirListBox.delete(0, END) # 폴더 리스트 박스를 비움
23         fileListBox.delete(0, END) # 파일 리스트 박스를 비움
24
25         dirListBox.insert(END, "상위폴더")
26         currentDir = searchDirList[len(searchDirList) - 1] # 디렉터리 리스트의 마지막이 현재 폴더.
27         dirLabel.configure(text = currentDir)
28         folderList = os.listdir(currentDir)
29         for item in folderList:
30             if os.path.isdir(currentDir + item) :
31                 dirListBox.insert(END, item)
32             elif os.path.isfile(currentDir + item) :
33                 fileListBox.insert(END, item)
```

368~370 페이지 응용예제 01 파일 탐색기 만들기 (4/8)

```
34
35     ## 전역 변수 선언 부분 ##
36     window = None
37     searchDirList = ['C:₩₩'] # 중요 변수! 검색한 폴더 목록의 스택
38     currentDir = 'C:₩₩'
39     dirLabel, dirListBox, fileListBox = None, None, None
40
41     ## 메인 코드 부분 ##
42     if __name__ == "__main__" :
43         window = Tk()
44         window.title("폴더 및 파일 목록 보기")
45         window.geometry('300x500')
46
47         dirLabel = Label(window, text = currentDir)
48         dirLabel.pack()
49
50         dirListBox = Listbox(window)
51         dirListBox.pack(side = LEFT, fill = BOTH, expand = 1)
52         dirListBox.bind('<<ListboxSelect>>', clickListBox)
53
54         fileListBox = Listbox(window)
55         fileListBox.pack(side = RIGHT, fill = BOTH, expand = 1)
56
57         fillListBox() # 초기엔 C:₩₩의 모든 폴더 목록을 만들기
58
59         window.mainloop()
```

368~370 페이지 응용예제 01 파일 탐색기 만들기 (5/8)

- 코드설명
 - 1~3행 관련 모듈을 임포트
 - 6~18행
 - clickListBox()함수의 작동원리를 대략적으로 먼저 살펴보자.
 - 이 함수는 윈도우창 왼쪽의 폴더 목록을 클릭했을때 실행된다.
 - 클릭한 폴더와 현재 폴더를 합쳐서 searchDirList에 추가한다.
 - 예를들어 현재 폴더가 C:WWtempWW 이고 클릭한 폴더가 program 폴더라면 searchDirList는 ["C: WW", "C:WW tempWW", "C: WWtempWWprogramWW"]이 된다.
 - 예외적으로 '상위폴더' 항목을 클릭했다면, searchDirList 의 마지막 항목을 삭제한다. 즉, searchDirList가 ["C:WW", "C:WWtempWW", "C:WWtempWWprogramWW"] 이었는데 '상위폴더'를 클릭하면, searchDirList가 ["C:WW", "C:WWtempWW"]가 된다.
 - searchDirList를 완성한 후 fillListBox()함수를 호출해서 화면의 두 리스트박스를 현재 폴더를 기준으로 다시 작성한다.
 - 8~9행은 다른 리스트박스를 클릭했을 때 발생하는 오류를 방지하기 위해서이다.
 - 10행에서 클릭한 폴더명을 문자열로 뽑는다.
 - 11~16행에서 클릭한 폴더가 '상위폴더'인데 현재 폴더가 "c:ww"일 때는 더 이상 상위폴더가 없으므로 무시한다. 그렇지 않으면 searchDirList에서 마지막 항목을 삭제한다. 결국 현재 폴더보다 상위폴더가 searchDirList의 마지막 항목이 된다.

368~370 페이지 응용예제 01 파일 탐색기 만들기 (6/8)

- 코드설명

- 20~33행

- 화면의 폴더 리스트박스과 파일 리스트박스를 새롭게 만든다.
 - 22~33행에서 우선 두 리스트박스를 깨끗하게 비운다. 핵심은 26행의 현재 폴더인 currentDir 인데, 현재 폴더는 항상 searchDirList의 마지막 항목을 사용한다.
 - 27행에서 위쪽 글자를 현재 폴더로 변경한다.
 - 28행에서 현재 폴더 안의 파일,폴더 목록을 뽑은 후
 - 29~33행에서 파일은 파일리스트박스에 삽입하고, 폴더는 폴더 리스트 박스에 삽입한다.

- 37행

- searchDirList가 이 프로그램에서 가장 중요한 변수이다. C:WW를 시작으로 검색한 목록이 기록되는 스택구조이다. C:WW 이후에 클릭한 항목들이 차곡차곡 쌓인다.
 - 예를 들어 C:WW 이후에 windows, system 폴더를 차례로 클릭했다면, searchDirList는 ["C:WW", "C:WWWindowsWW", "C:WWWindowsWWsystemWW"] 가 된다.
 - 이중 마지막 항목이 현재 폴더로 취급된다.
 - 그리고, 상위폴더로 이동하려고 마지막 항목을 삭제하면 자연스럽게 상위폴더가 다시 현재 폴더가 된다.

368~370 페이지 응용예제 01 파일 탐색기 만들기 (7/8)

- 코드설명

- 20~33행

- 화면의 폴더 리스트박스과 파일 리스트박스를 새롭게 만든다.
 - 22~33행에서 우선 두 리스트박스를 깨끗하게 비운다. 핵심은 26행의 현재 폴더인 currentDir 인데, 현재 폴더는 항상 searchDirList의 마지막 항목을 사용한다.
 - 27행에서 위쪽 글자를 현재 폴더로 변경한다.
 - 28행에서 현재 폴더 안의 파일,폴더 목록을 뽑은 후
 - 29~33행에서 파일은 파일리스트박스에 삽입하고, 폴더는 폴더 리스트 박스에 삽입한다.

- 37행

- searchDirList가 이 프로그램에서 가장 중요한 변수이다. C:WW를 시작으로 검색한 목록이 기록되는 스택구조이다. C:WW 이후에 클릭한 항목들이 차곡차곡 쌓인다.
 - 예를 들어 C:WW 이후에 windows, system 폴더를 차례로 클릭했다면, searchDirList는 ["C:WW", "C:WWWindowsWW", "C:WWWindowsWWsystemWW"] 가 된다.
 - 이중 마지막 항목이 현재 폴더로 취급된다.
 - 그리고, 상위폴더로 이동하려고 마지막 항목을 삭제하면 자연스럽게 상위폴더가 다시 현재 폴더가 된다.

368~370 페이지 응용예제 01 파일 탐색기 만들기 (8/8)

- 코드설명
 - 38행
 - 현재 폴더를 저장한 변수이다.
 - 항상 searchDirList의 마지막 항목을 뽑아서 저장할 것이다.
 - 39행
 - 윈도우 창에 나올 위젯변수들이다.
 - 40~47행
 - 위쪽 현재 폴더의 전체 경로를 출력한다.
 - 49~51행
 - 왼쪽 현재 폴더의 하위 폴더 목록을 보여주는 리스트박스이다.
 - 이 부분을 클릭하면 clickListBox() 함수를 호출해서 현재 폴더가 변경된다.
 - 53~54행
 - 왼쪽 현재 폴더의 하위 폴더 목록을 보여주는 리스트박스이다.
 - 이 부분을 클릭해도 아무런 작동을 하지 않는다.
 - 56행
 - 프로그램을 처음 실행하면 c:ww를 기준으로 화면을 구성해야 한다.
 - searchDirList의 초깃값이 ["C:WW"]이므로 자동으로 C:WW를 현재 폴더로 취급해서 화면을 구성한다.