

1월18일_1

[실습 1]

파일명 : fileioLab1.py

다음 내용을 생성해서 **c:/iotest/today.txt** 라는 파일에 저장한다.
c:/iotest 디렉토리의 존재여부를 체크하고 존재하지 않으면 새로이 생성한다.

출력 내용은 다음과 같습니다.

오늘은 **2021년 01월 18일**입니다.

오늘은 **월요일**입니다.

나는 **X요일**에 태어났습니다.

파일에 위의 내용을 저장한 다음에 화면에는**“저장이 완료되었습니다.”** 를 출력한다.

```
import os
import calendar
import datetime

now = datetime.datetime.now()
yoil = ['월', '화', '수', '목', '금', '토', '일']
day = calendar.weekday(1997, 9, 3)

if not os.path.isdir("c:/iotest"): # 존재하는가?
    os.mkdir("c:/iotest") # 없으면 만들어줘.

f = open("c:/iotest/today.txt", "wt", encoding="UTF-8")
f.write(f"오늘은 {now.year}년 {now.month:02d}월 {now.day:02d}일입니다.
오늘은 {yoil[calendar.weekday(now.year, now.month, now.day)]}요일입니다.
레크는 {yoil[day]} 요일에 태어났습니다.")
f.close()
print("저장이 완료되었습니다.")
```

```
import os
import time
import calendar

now = time.localtime()
week = ['월', '화', '수', '목', '금', '토', '일']

if not os.path.isdir("c:\\iotest"):
    os.mkdir("c:\\iotest")

f = open("c:\\iotest\\today.txt", "wt", encoding="UTF-8" )

f.write(f"오늘은 {now.tm_year}년 {now.tm_mon:02d}월 {now.tm_mday:02d}일입니다.
오늘은 {week[calendar.weekday(now.tm_year, now.tm_mon, now.tm_mday)]}요일입니다.
나는 {week[calendar.weekday(1997, 10, 17)]}요일에 태어났습니다.")

f.close()
```

```
print("저장이 완료되었습니다.")
```

[실습 2]

파일명 : fileioLab2.py

제공된 sample.txt 파일을 읽고 sample_yyyy_mm_dd.txt 파일에 그대로 출력하는 프로그램을 구현한다.

이 파일은 **append 모드로 오픈**하여 여러 번 테스트하면 sample.txt 파일의 내용이 sample_yyyy_mm_dd.txt 파일에 계속 추가되게 한다.

위의 내용을 파일에 저장하는 기능이 정상적으로 수행되면 화면에 “저장이 완료되었습니다.”

FileNotFoundError 발생시 **FileNotFoundError** 에 대한 메시지 정보를 출력하고 종료한다.

```
import time

now = time.localtime()
try:
    fr = open("C:/jsj/PYTHONexam/day10/sample.txt", "rt", encoding="UTF-8")
    file_name = f"sample_{now.tm_year}_{now.tm_mon:02d}_{now.tm_mday:02d}.txt"
    fa = open(file_name, "at", encoding="UTF-8")

    for line in fr:
        fa.write(line)

    fr.close()
    fa.close()

except FileNotFoundError as e:
    print(e)
print("저장이 완료되었습니다.")
```

```
import time

now = time.localtime()
try:
    fr = open("sample.txt", "rt", encoding="UTF-8")

    file_name = f"sample_{now.tm_year}_{now.tm_mon:02d}_{now.tm_mday:02d}.txt"
    fw = open(file_name, "at", encoding="UTF-8")

    for line in fr: # _io.TextIOWrapper 객체도 리터러블함
        fw.write(line)

    fr.close()
    fw.close()
except FileNotFoundError as e:
    print(e)
```

[실습 3]

파일명 : stLab1.py

프로그램 수행 중간에 **년도와 월을 입력받아** 해당년과 월의 달력을 출력한다.

년도와 월은 **숫자로** 입력받아야 하며 숫자가 입력되지 않아서 발생하는 에러를 대비해야한다.

ValueError 처리를 통해서 숫자가 입력되지 않은 경우에는 메시지를 내보내고 **다시 입력받는다**. 잘 할 때까지.....

년도와 월이 제대로 입력되면 **해당 년월의 달력을 출력한다**.

```
import calendar
while True:
    year = input("년도를 입력하세요 : ")
    month = input("월을 입력하세요 : ")
    try:
        x = int(year)
        y = int(month)
        calendar.prmonth(x, y)
        break
    except ValueError:
        print("숫자로 다시 입력해주세요")
```

```
import calendar

while True:
    try:
        year = int(input("년도를 입력하시오 :"))
        mon = int(input("월을 입력하시오 :"))
        print(calendar.month(year, mon))
        break
    except ValueError:
        print("숫자로 다시 입력해주세요")
```

```
import calendar

while True:
    str_year = input('년도를 입력하세요 : ')
    str_month = input('월 입력하세요 : ')
    try:
        int_year = int(str_year)
        int_month = int(str_month)
        print(calendar.month(int_year, int_month))
        break
    except ValueError:
        print("숫자로 다시 입력해주세요")
```

```
import calendar

while True:
    str1 = input("년도를 입력하세요 : ")
    str2 = input("월을 입력하세요 : ")
    try:
        year = int(str1)
        month = int(str2)
```

```

        if month not in range(1,13):
            raise ValueError("월은 1~12 사이 입력요!!")
        if year < 0 :
            raise ValueError("년도는 양의 값으로 입력해요!!")
        break
    except ValueError as v:
        print("입력값이 잘못되었습니다.(" + str(v) +)")

print(calendar.month(year, month))

```

[실습 4]

파일명 : fileioLab3.py

제공된 yesterday.txt 파일을 읽고 이 문서에** "Yesterday" 가 몇 개 들어 있는지 개수를 세어서 출력** 한다.

(이 때 대소문자는 구분하지 않는다.)

FileNotFoundError 발생시 파일을 읽을 수 없어요 를 출력하고 종료한다.

에러가 발생하지 않으면 **Number of a Word 'Yesterday' X** 를 출력하고 종료한다.

(참고 : X는 9인걸루 예측됨)

에러 발생 여부와 관계없이 수행완료!! 를 출력하고 종료한다.

(try-except-else-finally 를 모두 사용해서 해결한다.)

```

import os
f = None
try:
    f = open("yesterday.txt", "rt", encoding="UTF-8")
    text = f.read()
    max = text.lower()
    c = max.count("yesterday")
except FileNotFoundError:
    print("파일을 읽을 수 없어요.")
else:
    print("Number of a word 'yesterday'", c)
finally:
    print("수행완료!!")
    if f :
        f.close()

```

```

cnt = 0
try:
    f = open("yesterday.txt", "rt", encoding="UTF-8")
except FileNotFoundError :
    print("파일을 읽을 수 없어요")
else:
    for tmp in f:
        tmp = tmp.lower()
        cnt += tmp.count('yesterday')
    print(f"Number of a Word 'Yesterday' {cnt}")
finally:
    print("수행완료!!")

```

```

f = None
try :
    f = open("yesterday1.txt", "rt", encoding="UTF-8")
    text = f.read()
    yesterday_count=text.count("yesterday")+text.count("Yesterday")
    print(yesterday_count)
except FileNotFoundError:
    print("파일을 읽을 수 없어요")
else :
    print("Number of a word 'Yesterday'", yesterday_count)
finally :
    print("수행완료!")
    if f :
        f.close()

```

```

try:
    with open("yesterday.txt", "rt") as f:
        text = f.read()
        text = text.lower()
        number = text.count('yesterday')
        #f.close()
except FileNotFoundError:
    print('파일을 읽을 수 없어요.')
else:
    print("Number of a word 'Yesterday'+str(number))
finally:
    print('수행완료!!')

```