

파일 입출력

파일 생성

```
#-----  
# 1. 파일 생성  
#-----  
# open(파일경로, 모드)  
# 모드 : r(읽기) w(쓰기) a(추가)  
# 리눅스&유닉스(슬러시) : /etc/guest/aa.txt  
# 윈도우(역슬러시) : C:\aaa\bb.txt  
# 파이썬 프로그래밍 : C:\\aaa\\bb.txt or C:/aaa/bb.txt  
  
f =  
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_ansi.txt"  
      , mode='r')  
f.close()
```

파일 읽기

```
#읽기  
#ANSI ASCII  
#CP949 UTF-8  
#-----  
#readline() : 라인 1줄  
#-----  
  
f =  
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_ansi.txt"  
      , mode='r')  
txt=f.readline()  
print(txt)  
f.close()  
  
f =  
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_utf8.txt"  
      , mode='r'  
      , encoding="utf-8")  
txt=f.readline()  
print(txt)  
f.close()
```

```

f =
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_utf8.txt"
    , mode='r'
    ,encoding="utf-8")

while(True): #조건이 만족하면 while 가동
    txt=f.readline()
    if not txt: #txt에 더이상 읽을 것이 없으면
        break
    print(txt, end="")
f.close()

print("\n", "--"*40)

#-----
# read(), readlines() :전체
# 리스트로 반환
#-----
f =
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_utf8.txt"
    , mode='r'
    ,encoding="utf-8")

txt_list =f.readlines() #['동해물과 \n', '백두산이 \n', '마르고 닳도록']
for txt in txt_list:
    txt=txt.strip() #공백제거&개행(줄바꿈)제거
    print(txt, end="")
f.close()

print("\n", "--"*40)

f =
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_utf8.txt"
    , mode='r'
    ,encoding="utf-8")

txts = f.read()
print(txts)
f.close()

#-----
# with open() as f : 들여쓰기 라인이 모두 실행 되면 자동으로 close()
#-----
with
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_utf8.txt"
    , mode='r'
    ,encoding="utf-8") as f:
    txts = f.read()
    print(txts)

```

파일 쓰기

```
#-----
#파일 쓰기
#with open(file="파일경로", mode='w') as f :
#-----

with
open("C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonPr
ject\\venv\\test\\memo_write.txt",mode='w') as f :
    f.write(str(1)+"\n")
    f.write("2\n")
print("===done===")

with
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pyt
honProject\\venv\\test\\memo_write.txt",mode='w') as f:
    for i in range(1, 11):
        f.write(str(i) + "\n")
print("===done===")
```

파일 복사

```
#-----
#파일 복사(copy)
#memo_utf8.txt 파일을 memo_write.txt에 복사
#-----

fr=open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\
pythonProject\\venv\\test\\memo_utf8.txt",encoding='utf-8',mode='r')
fw=open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\
pythonProject\\venv\\test\\memo_write.txt",encoding='utf-8',mode='a')

txt_list=fr.readlines()
for txt in txt_list:
    fw.write(txt)
fr.close()
fw.close()

print("==== copy done====")
with
open(file="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pyt
honProject\\venv\\test\\memo_write.txt"
    , encoding='utf-8',mode='r') as f:
    print(f.read())

#-----
from shutil import copy,copy2,copyfile
```

```
#-----
# copy(원본파일경로, 타겟파일경로)
# copy ("/etc/memo/a.txt", "/etc/memo2/b.txt")
# copy2("/etc/memo/a.txt", "/etc/memo2") --> /etc/memo2/a.txt
#-----
orig="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_utf8.txt"
dest="C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\memo_write.txt"

copyfile(orig,dest)
```

파일 입출력 함수

```
#-----
# 패키지   모듈       클래스   함수
# 폴더     .py파일   class  def
#-----
# 함수     : 공통의 기능을 정의해 놓음
# 클래스   : 함수들을 관리하기 위한 큰 틀

class CacClass :
    def add(n1,n2):
        res = n1 + n2
        return res

class MemberClass:
    def add(name, age):
        print(name,"저장")

def myprint(n):
    print(n,"입력되었습니다")

def uprint(n):
    print(n,"입력되었습니다2222")
#-----
# 함수를 불러서 사용한다 --> 함수 호출
# 함수에 return 이 있는 경우 변수에 바인딩에 사용할 수 있다.
#-----
a= CacClass.add(1,5)
print(a)
print(CacClass.add(1,5))

MemberClass.add("kim",30)

myprint(5)
```

파일 입출력 함수 호출

```
#-----
```

```

# 함수를 불러서 사용한다 --> 함수 호출
# 함수에 return 이 있는 경우 변수에 바인딩에 사용할 수 있다.
#-----
#PycharmProjects.pythonProject.venv.test.1ec04_파일입출력함수
#-----
# 1. from : 패키지명.모듈명
# 2. import : 클래스명, 모듈명, 함수명
# 3. 모듈명은 일반적으로 별칭(alias)를 사용한다
# 4. import 함수명1, 함수명2 이 길어지면 import 모듈명으로 한다
#
#     ex) from 패키지명.모듈명 import 클래스명
#     ex) from 패키지명.모듈명 import 함수명1, 함수명2 ...
#     ex) from 패키지명 import 모듈명 as 별명
#-----
from PycharmProjects.pythonProject.venv.test.1ec04_파일입출력함수 import CacClass
from PycharmProjects.pythonProject.venv.test.1ec04_파일입출력함수 import
MemberClass
from PycharmProjects.pythonProject.venv.test.1ec04_파일입출력함수 import myprint,
uprint

a= CacClass.add(1,5)
print(a)
print(CacClass.add(1,5))

MemberClass.add("kim", 30)
myprint(5)
uprint(5)

#-----
from PycharmProjects.pythonProject.venv.test import 1ec04_파일입출력함수
#-----
1ec04_파일입출력함수.uprint(55)
1ec04_파일입출력함수.MemberClass.add("park", 29) #----비추

#-----
from PycharmProjects.pythonProject.venv.test import 1ec04_파일입출력함수 as lkh
#-----
lkh.uprint(55)

```

파일 & 디렉토리 내장 함수

```

#-----
#-----
# 파일&디렉토리 내장 함수
# 절대 경로(Absolute Path) :
C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\
\\venv\\test
# 상대 경로(Relative Path) : 상대적으로 내가 위치한 폴더를 기준으로하는 경로
#     . : 현재폴더
#     .. : 상위폴더
#     / : 아래
#-----

```

```

#-----
import os
from os.path import isfile, isdir, join

print(os.getcwd()) # os.getcwd():파일 경로 출력

print(os.listdir("C:\\Users\\ASIA\\PycharmProjects\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test"))
print(os.listdir(os.getcwd())) # os.listdir(): 현재 위치의 모든 디렉토리 출력

pwd = os.getcwd()
print(os.listdir(pwd))

print("\n","--"*40)
#-----
#os.listdir(pwd)로 출력되는 디렉토리(폴더)들에 파일 경로 붙이기
for f in os.listdir(pwd):
    print(join (os.getcwd(),f))

print( join ("/aa","\bb.txt")) #경로 합치기

print("\n","--"*40)

#-----
#-----
#os.listdir() : 현재 위치의 모든 디렉토리 출력
# os.isdir() : 해당 디렉토리 존재 여부
# os.isfile() : 해당 파일 존재 여부
# os.path.join : 출력되는 디렉토리(폴더)들에 파일 경로 붙이기
#-----
#-----번외
for f in os.listdir("C:\\") :
    path = join( "C:\\", f) # join 걸지 않으면 현재 위치에서 파일을 찾을
    if os.path.isdir(path) : #isdir() : 해당 디렉토리 존재 여부
        if f == "AI" :
            subpath = join("C:\\", f) #C:\\AI
            for sf in os.listdir(subpath):
                fullname = join(subpath, sf) #C:\\AI\\python38
                if os.path.isdir(fullname):
                    print("-----", sf)
                elif os.path.isfile(fullname):
                    print("-----", sf)
            else :
                print("[D]", f)
        elif os.path.isfile(path) : #isfile() : 해당 파일 존재 여부
            print("\t", f)

#-----
#-----
# getmtime() : 수정 시간 구하기
# fromtimestamp() : 시간형태로 변환
# strftime() : 지정 형식으로 시간 변환
#-----
#-----
#datetime.py
# class date

```

```

# class datetime
# class datedelta
# class ...
#-----
#from 패키지(폴더)/datetime.py import strftime()
#from datetime import datetime
#datetime.strftime()

#from datetime.datetime import strftime
#strftime(datetime타입, "포맷")

#datetime.datetime.strftime()
# 모듈    클래스    함수
#-----
import datetime as dt

float_mm = os.path.getmtime("C://Setup.log")    #1541654654.456464465
print( type(float_mm),    float_mm)

time_mm = dt.datetime.fromtimestamp(float_mm) #시간형태로 변환
print( type(time_mm),    time_mm)

str_mm = dt.datetime.strftime(time_mm, "%Y-%m-%d %H:%M:%S")
print( type(str_mm),    str_mm )

#-----
for f in os.listdir("C:\\") :
    path = join( "C:\\", f)

    # -----수정날짜 구하기-----
    float_mm = os.path.getmtime("C://Setup.log")    #1535692326.6707716
    time_mm = dt.datetime.fromtimestamp(float_mm)    # 시간형태로 변환 #2018-08-31
14:12:06.670772
    str_mm = dt.datetime.strftime(time_mm, "%Y-%m-%d %H:%M:%S") #2018-08-31
14:12:06
    # -----
    if os.path.isdir(path):
        print(f, "\\t\\t\\t", "파일폴더", "\\t\\t", str_mm)
    elif os.path.isfile(path):
        fsize =os.path.getsize(path) #파일 사이즈 출력(기본 byte 출력)
        print(f, "\\t\\t\\t", "텍스트문서", "\\t\\t", str_mm, "\\t\\t", fsize/1024, "KB")
            #(byte*1024=kbyte)xxxxxxxxxx

#-----
#-----glob.py
# /etc/*.txt
# /etc/*c/a*.log
#-----
#-----
import glob

print( glob.glob("C:\\P*") )

print( glob.glob("C:\\P*\\M*") )
print( glob.glob("C:\\AI\\pythonProject\\venv\\python-basic\\lec*.py") )

```

```
print("-----")
glist = glob.glob("C:\\AI\\pythonProject\\venv\\python_basic\\lec*.py")
module_list = []
for fname_str in glist:
    module_list.append( fname_str.split("\\")[-1] )
    # print(os.path.split(fname_str))      #(경로, 파일명)
    # print(os.path.basename(fname_str))    #파일명
    # print(os.path.dirname(fname_str))     #경로
    # print(os.path.splitext(fname_str))    #( 경로/파일명 , .py(확장자) )
print( module_list )
```