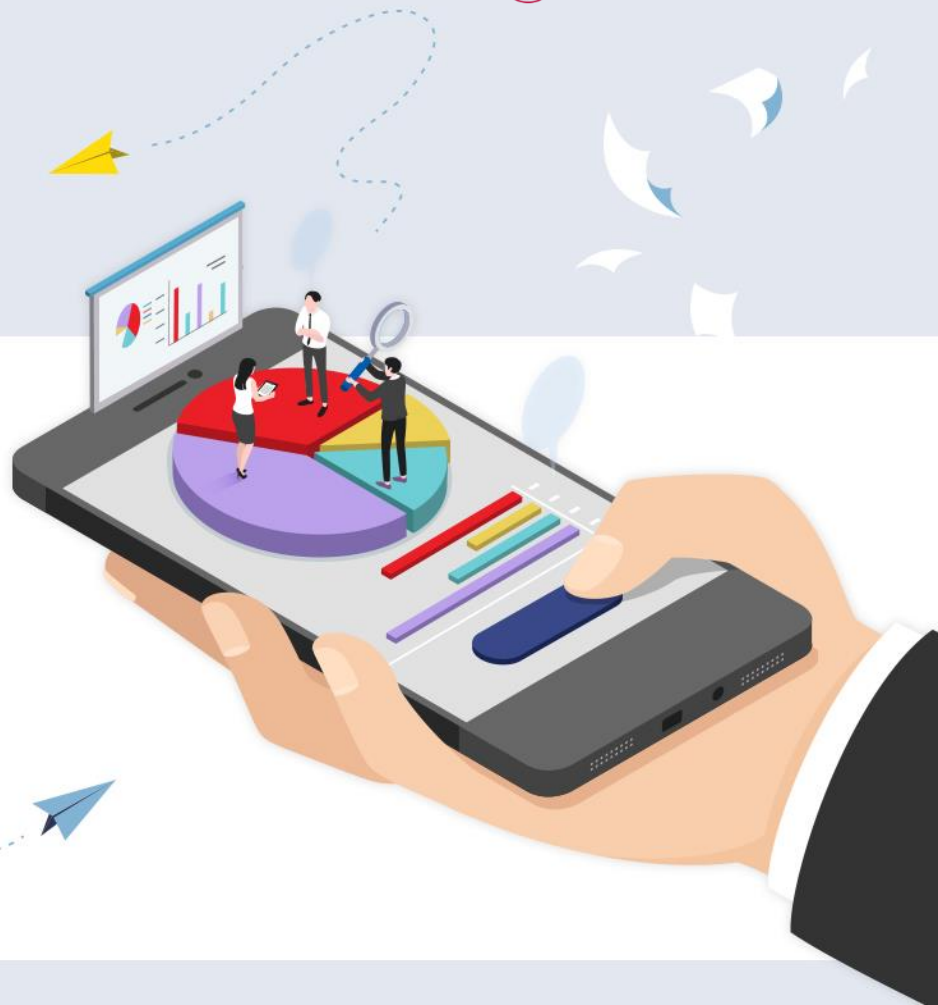




01

파일 입·출력의 기본

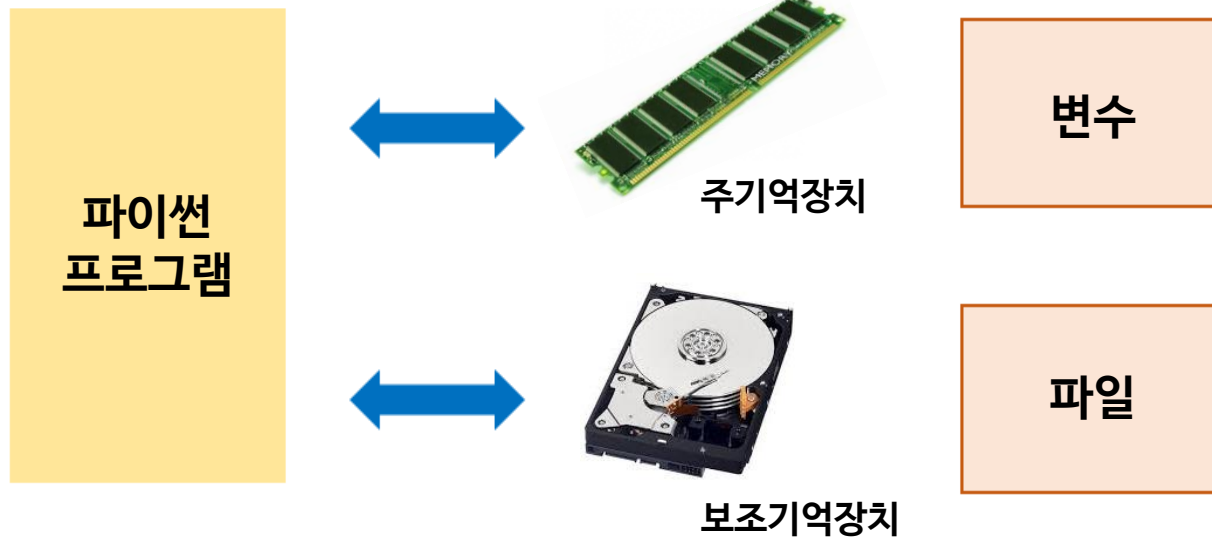


A diagram on the left side of the slide. It features a large, solid purple semi-circle. To its right, a thin purple arc with two small white circles at its ends connects the semi-circle to the text blocks. The word '파일' is written in white inside the purple semi-circle.

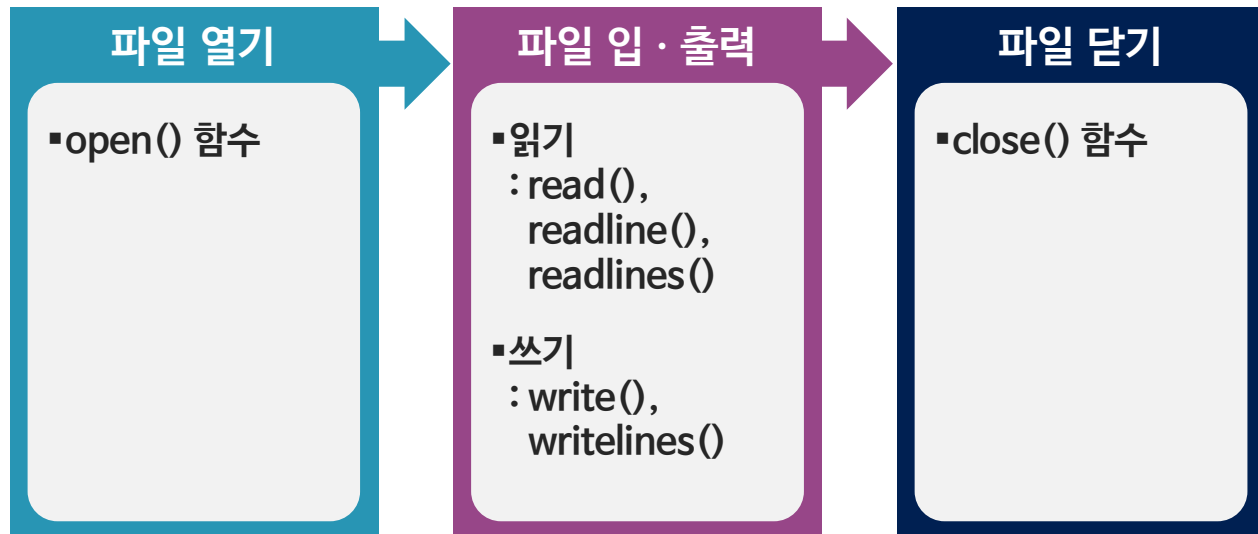
파일

실행 중인 프로그램이 종료되면
처리한 결과값은 메모리에서 사라지기 때문에,
모든 데이터는 더 이상 사용할 수가 없음

프로그램을 실행하는 도중에 데이터를 영구적으로
저장하고 싶다면, 보조기억장치(하드디스크)에
파일 형태로 저장해야 함

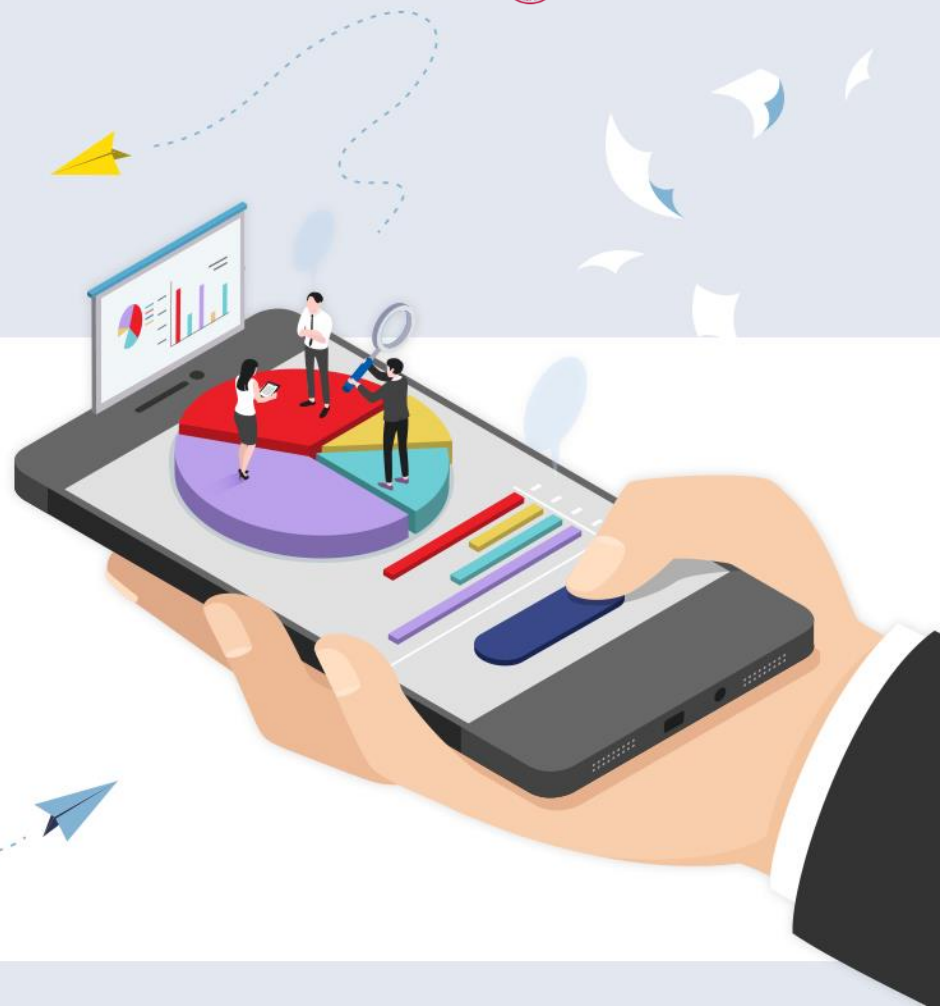


2 파일 입 · 출력의 기본 과정





02 텍스트 파일 읽고 쓰기



✓ 디렉토리(파이썬 파일이 저장된 위치)에 있는 파일을 열고
파일을 생성하기 위해 파이썬 내장 함수 `open()`를 사용

- `open()` 함수

: “파일 이름”과 “파일 열기 모드”를 입력값으로 받고
결과값으로 파일 객체를 반환하는 함수

✓ 파일 객체 = open(파일명, 파일열기모드)

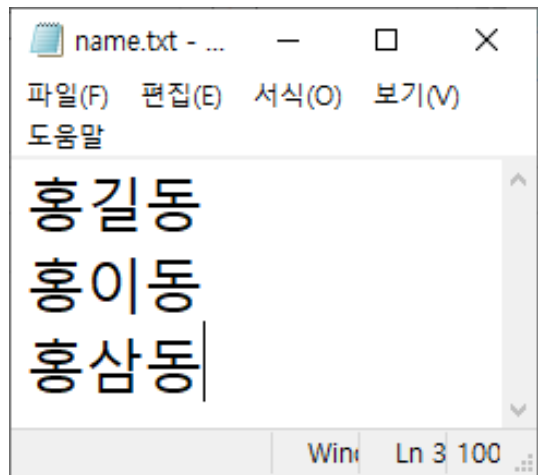
- 읽기 : 변수명 = open('파일명', 'r')
- 쓰기 : 변수명 = open('파일명', 'w')

✓ 파일 열기 모드

종류	설명
r	읽기모드 (Read), 기본값
w	쓰기모드 (Write)
a	추가모드 (Append)

외부의 파일을 읽어서 프로그램에서 사용할 수 있는
여러 가지 방법이 있는 파이썬

데이터 읽기를 실행하기위해 필요한 텍스트 파일



디렉토리에
'name.txt' 파일을 생성하여
여러 가지로 방법으로
파일 내용을 읽어서 출력



예1 파일 읽기 : read() 함수

```
f=open('name.txt')  
data=f.read()  
print(data)  
f.close()
```

>>>

홍길동

홍이동

홍삼동

- read() 함수
: 파일의 전체 내용을 문자열로 반환함



예2 파일 읽기 : readline() 함수

```
f=open('name.txt')  
data=f.readline()  
print(data)  
f.close()
```

실행결과

>>>

홍길동

- readline() 함수
: 파일의 내용을 한 줄씩 읽어서
문자열로 반환함



예3 파일 읽기 : readline() 함수

```
f=open('name.txt')
while True:
    data=f.readline()
    if not data:break
    print(data,end='')
f.close()
```

>>>

홍길동

홍이동

홍삼동



예4 파일 읽기 : readlines() 함수

```
f=open('name.txt')  
data=f.readlines()  
print(data,end='')  
f.close()
```

```
>>>  
['홍길동\n', '홍이동\n', '홍삼동']
```

- readlines() 함수
: 파일의 모든 줄을 읽어서
리스트로 반환함
- readline() 함수
: 한 줄씩 읽음

예5 파일 읽기 : readlines() 함수

```
f=open('name.txt')
data=f.readlines()
for line in data:
    print(line,end='')
f.close()
```

>>>

홍길동

홍이동

홍삼동

- 파이썬에서는 주로 with문을 사용하여 파일 처리를 함
- with 블록을 벗어나면 자동으로 파일이 닫힘(자동 close)

```
with open('name.txt') as f:
    data=f.readlines()
    print(data,end='')
```

- ✓ 프로그램에서 처리한 결과 값을 파일에 저장하기 위해서는 `open()` 함수로 파일을 쓰기모드('w')로 생성한 후, **다양한 함수** 이용



`write()`
`writelines()`



예6

파일 쓰기 : write() 함수

- 동아리 회원의 이름을 사용자로부터 입력 받아 파일에 저장해보자.

```
f=open('club.txt','w')
```

```
while True:
```

```
    name=input('name:')
```

```
    if not name:break
```

```
    f.write(name)
```

```
f.close()
```

```
>>>
```

```
name:황채연
```

```
name:황나연
```

```
name:황복동
```

```
name:
```


예6 파일 쓰기 : write() 함수

- 동아리 회원의 이름을 사용자로부터 입력 받아 파일에 저장해보자.

```
f=open('club.txt','w')
while True:
    name=input('name:')
    if not name:break
    f.write(name)
f.close()
```

```
>>>
```

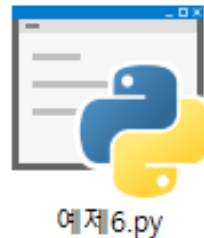
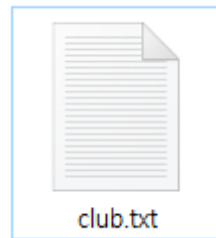
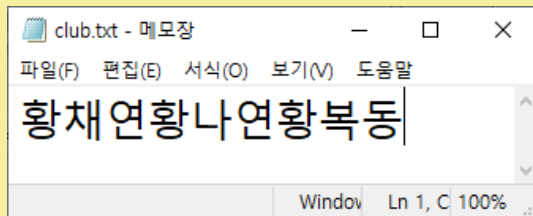
```
name:황채연
```

```
name:황나연
```

```
name:황복동
```

```
name:
```

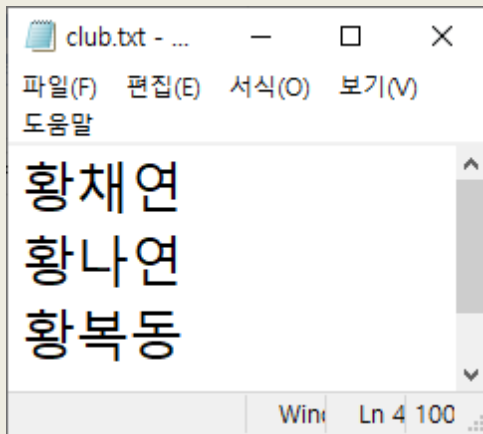
- 프로그래를 실행시킨 후, 현재 디렉토리를 살펴보면 'club.txt' 파일이 생성되었음을 확인하고 파일을 열어 입력 받은 내용이 저장되었는지를 확인해보기



- ✓ write() 함수는 자동 개행 기능이 포함되어 있지 않기 때문에 파일에 줄 단위로 저장하고 싶다면 개행 문자('\n')를 포함시켜주면 됨

예7

```
f=open('club.txt','w')
while True:
    name=input('name:')
    if not name:break
    f.write(name+'\n')
f.close()
```



- ☑ 프로그램을 실행시킨 후 'club.txt' 파일이 생성되었음을 확인하고, 파일을 열어 입력 받은 내용이 파일 맨 뒤에 추가 되어 저장되었는지를 확인해볼 것

예8

파일 추가모드 : write() 함수

- 동아리 회원의 이름을 사용자로부터 입력 받아
기존 파일(club.txt)에 추가해보자.

```
f=open('club.txt','a')
while True:
    name=input('name:')
    if not name:break
    f.write(name+'\n')
f.close()

>>>
name:정재경
name:정태경
name:
```

- 파일 쓰기 모드('w')로 파일을 열 때는 기존 파일이 있을 경우 그 파일의 내용이 모두 사라짐
- 기존 파일 내용을 유지하면서 단지 새로운 값을 추가해야 하는 경우에는 파일을 추가모드('a')로 생성하면 됨

예8

파일 추가모드 : write() 함수

- 동아리 회원의 이름을 사용자로부터 입력 받아
기존 파일(club.txt)에 추가해보자.

