

**산업 인공지능 - 실습 8**

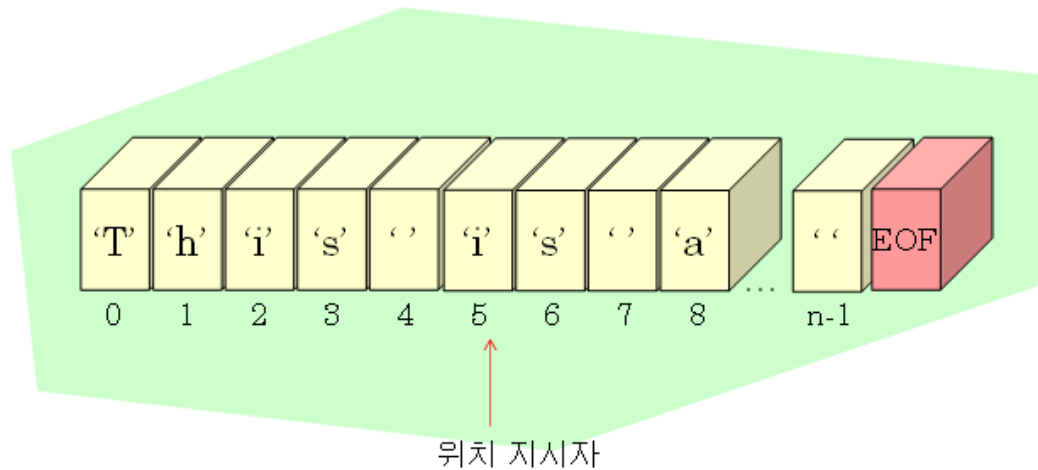
# **Python 프로그래밍**

2021 Spring

# 1. 파일

## ❖ 논리적 파일의 구조

- 파일 안에는 바이트들이 순차적으로 저장
- 끝에는 EOF(end-of-file) 마커 존재

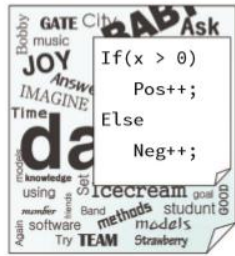


파일의 논리적인 구성

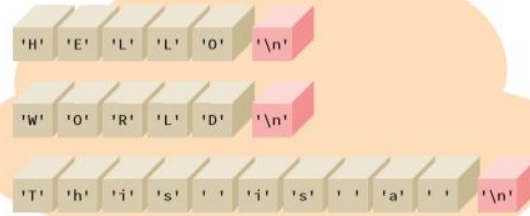
# 파일

## ❖ 파일의 종류

- 텍스트 파일(text file)



텍스트 파일: 문자로 구성된 파일

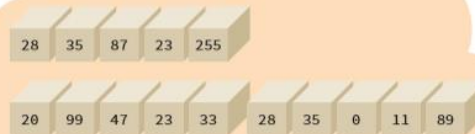


텍스트 파일

- 이진 파일(binary file)



이진파일: 데이터로 구성된 파일

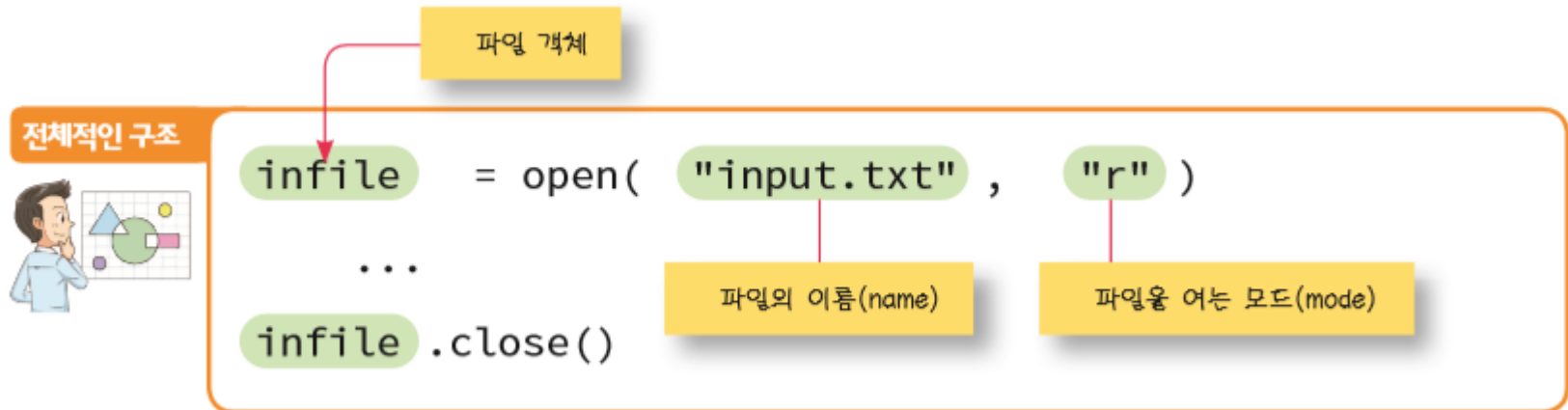


이진 파일

# 텍스트 파일

## ❖ 파일 열기와 닫기

- 쓰거나 읽기 위해 먼저 열어야 함
- 사용 후에는 닫아야 함



```
infile = open("phones.txt", "r")  
s = infile.read(10)  
print(s)  
infile.close()
```

# 텍스트 파일

## ❖ 파일 모드(mode)

- 여는 파일의 사용 용도 지정

파일 모드	모드 이름	설명
"r"	읽기 모드(read mode)	파일의 처음부터 읽는다.
"w"	쓰기 모드(write mode)	파일의 처음부터 쓴다. 파일이 없으면 생성된다. 만약 파일이 존재하면 기존의 내용은 지워진다.
"a"	추가 모드(append mode)	파일의 끝에 쓴다. 파일이 없으면 생성된다.
"r+"	읽기와 쓰기 모드	파일에 읽고 쓸 수 있는 모드이다. 모드를 변경하려면 seek()가 호출되어야 한다.



"r"

파일을 처음부터 읽는다.



"w"

파일의 처음부터 쓴다.  
만약 파일이 존재하면  
기존의 내용이 지워진다.



"a"

파일의 끝에 쓴다.  
파일이 없으면 생성된다.

```
infile = open("phones.txt", "r")
s = infile.read(10)
print(s);
infile.close()
```

# 텍스트 파일

## ❖ 파일 읽기

```
infile = open("phones.txt", "r")
line = infile.readline()
while line != "":
    print(line)
    line = infile.readline()
infile.close()
```

```
홍길동 010-1234-5678
김철수 010-1234-5679
김영희 010-1234-5680
```

# 텍스트 파일

## ❖ 파일에 쓰기

```
import os.path

outfile = open("phones.txt", "w")

if os.path.isfile("phones.txt"):
    print("동일한 이름의 파일이 이미 존재합니다. ")
else :
    outfile.write("홍길동 010-1234-5678")
    outfile.write("김철수 010-1234-5679")
    outfile.write("김영희 010-1234-5680")

outfile.close()
```

# 텍스트 파일

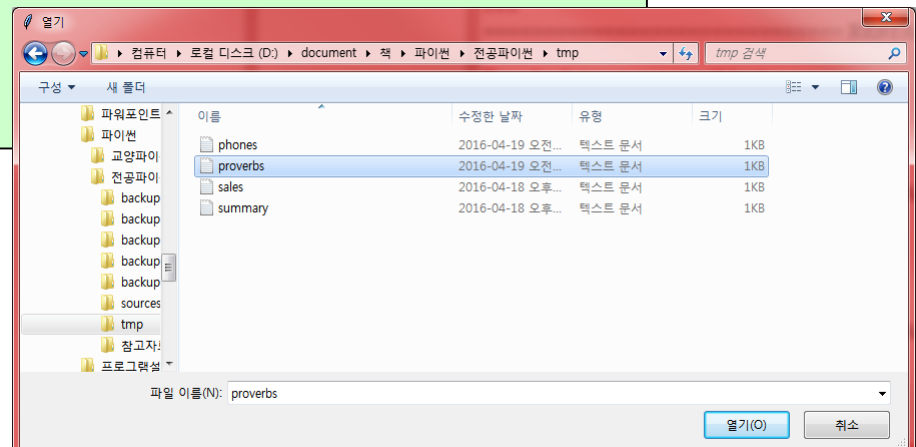
## ❖ 파일 대화 상자

```
from tkinter import *
from tkinter.filedialog import askopenfilename
from tkinter.filedialog import asksaveasfilename

readFile = askopenfilename()
If readFile != None:
    infile = open(readFile, "r")

for line in infile.readlines():
    line = line.strip()
    print(line)

infile.close()
```

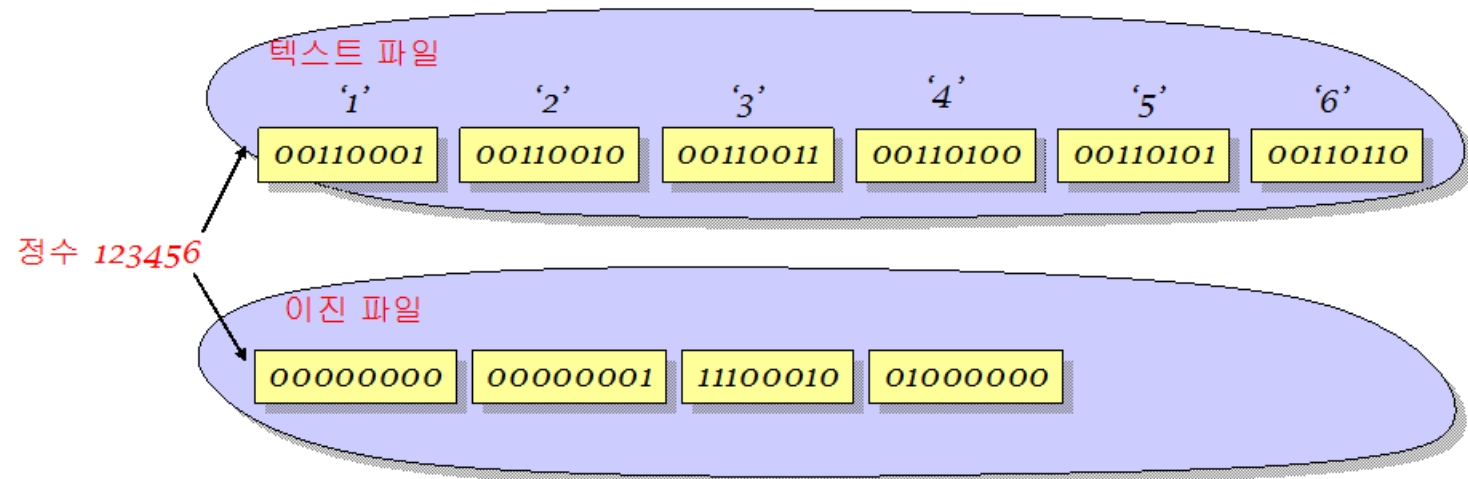




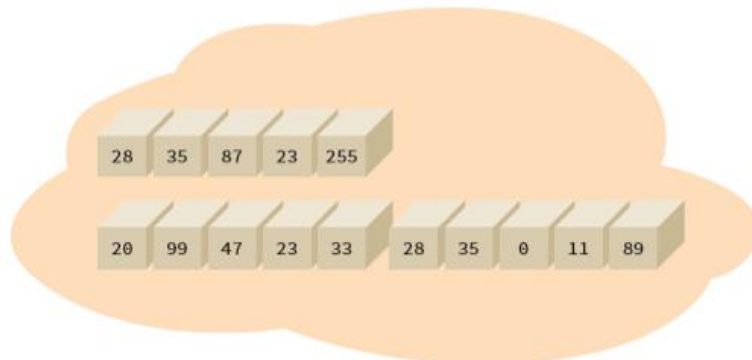
# 이진 파일

## ❖ 이진 파일(binary file)

- 텍스트가 아닌 데이터를 저장



이진파일: 데이터로 구성된 파일



이진 파일

# 이진 파일

```
filename1 = input("원본 파일 이름을 입력하시오: ");
filename2 = input("복사 파일 이름을 입력하시오: ");

infile = open(filename1, "rb")
outfile = open(filename2, "wb")

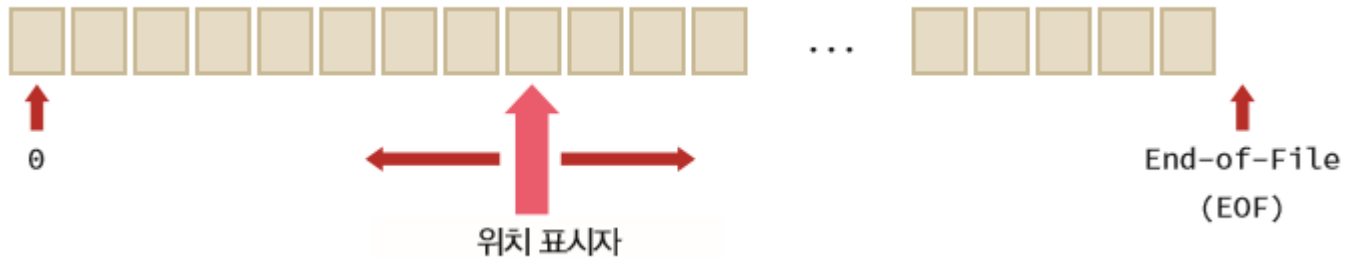
# 입력 파일에서 1024 바이트씩 읽어서 출력 파일에 쓴다.
while True:
    copy_buffer = infile.read(1024)
    if not copy_buffer:
        break
    outfile.write(copy_buffer)

infile.close()
outfile.close()
print(filename1+"를 " +filename2+"로 복사하였습니다. ")
```

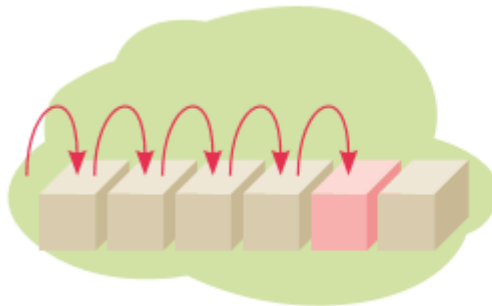
# 파일의 특정 위치 접근

## ❖ 위치 표시자

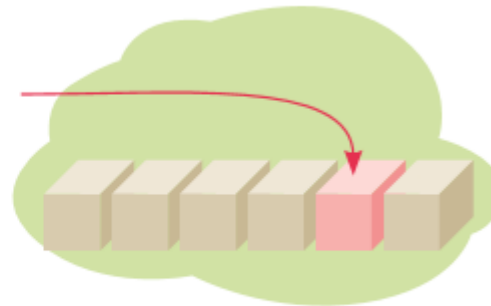
- 파일을 읽거나 쓰고 있는 위치 지시



- 순차접근과 임의접근



순차접근파일



임의접근파일

# 파일의 특정 위치 접근

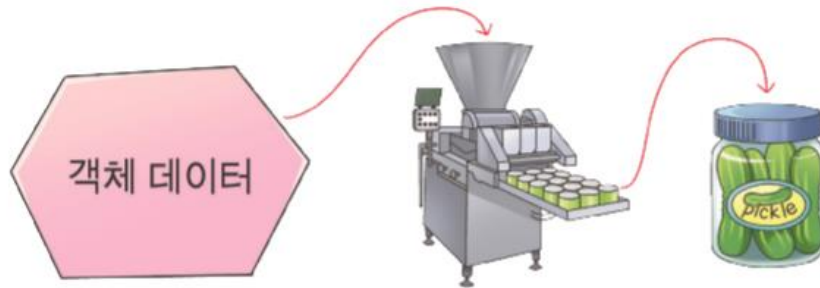
```
infile = open("test.txt", "r+")  
str = infile.read(10);  
print("읽은 문자열 : ", str)  
position = infile.tell();  
print("현재 위치: ", position)  
  
position = infile.seek(0, 0);  
str = infile.read(10);  
print("다시 읽은 문자열 : ", str)  
infile.close()
```

읽은 문자열 : All's well  
현재 위치: 10  
다시 읽은 문자열 : All's wel

# 객체의 파일 입출력

## ❖ 객체의 파일 저장 및 읽어오기

- pickle 모듈의 `dump()`와 `load()` 메소드 사용



```
import pickle
myMovie = {"Superman vs Batman ": 9.8, "Ironman": "9.6"}

# 딕셔너리를 피클 파일에 저장
pickle.dump(myMovie, open("save.p", "wb"))

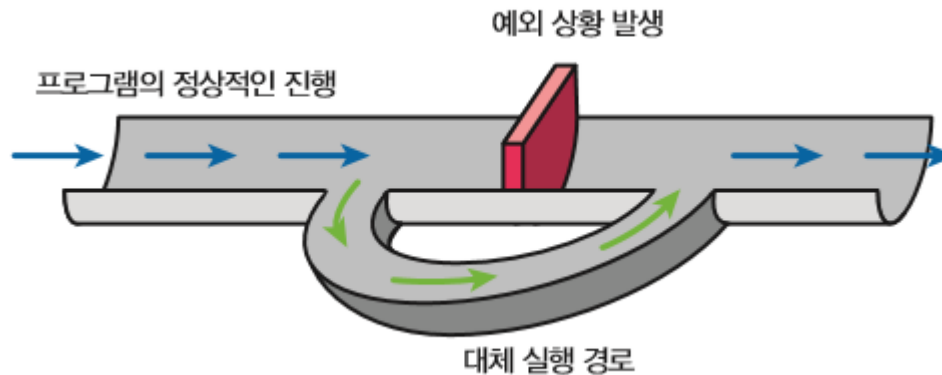
# 피클 파일에 딕셔너리를 로딩
myMovie = pickle.load(open("save.p", "rb"))
print(myMovie)
```

```
{'Superman vs Batman ': 9.8, 'Ironman': '9.6'}
```

## 2. 예외처리

### ❖ 예외처리(exception handling)

- 오류 발생시 프로그램이 바로 종료되지 않고, 프로그램에서 오류를 처리하고 종료할 수 있도록 하는 기능



```
>>> (x, y)=(2, 0)
>>> z=x/y
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    z=x/y
ZeroDivisionError: division by zero
>>>
```

# 예외처리

## ❖ 예외처리 구조

전체적인 구조



try :

예외가 발생할 수 있는 문장

except 오류내용 :

예외를 처리하는 문장

```
(x,y) = (2,0)
try:
    z = x/y
except ZeroDivisionError:
    print ("0으로 나누는 예외")
```

0으로 나누는 예외

# 예외처리

```
while True:
    try:
        n = input("숫자를 입력하시오 : ")
        n = int(n)
        break
    except ValueError:
        print("정수가 아닙니다. 다시 입력하시오. ")
print("정수 입력이 성공하였습니다!")
```

숫자를 입력하시오 : 23.5  
정수가 아닙니다. 다시 입력하시오.  
숫자를 입력하시오 : 10  
정수 입력이 성공하였습니다!



# 예외처리

## ❖ 다중 예외 처리 구조

전체적인 구조



try :

예외가 발생할 수 있는 문장

except ExceptionI :

ExceptionI이면 이 블록이 실행된다.

except ExceptionII :

ExceptionII이면 이 블록이 실행된다.

else :

예외가 없는 경우에 실행된다.

# 예외처리

```
try:
    fh = open("testfile", "w")
    fh.write("테스트 데이터를 파일에 씁니다!!")
except IOError:
    print("Error: 파일을 찾을 수 없거나 데이터를 쓸 수 없습니다. ")
else:
    print("파일에 성공적으로 기록하였습니다. ")
    fh.close()
```

*파일에 성공적으로 기록하였습니다.*