LESSON

파일 입출력

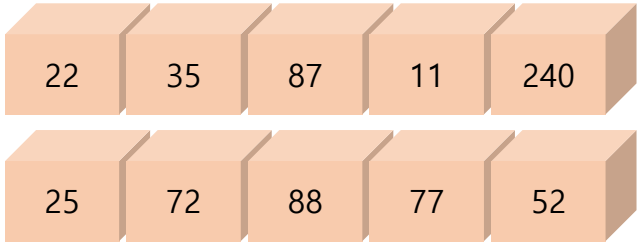
파일의 종류

텍스트 파일(text file)

- 사람이 읽을 수 있는 텍스트가 들어 있는 파일.
- 파일에는 문자들이 들어 있고 이들 문자들은 아스키 코드를 이용해서 표현된다.
- 텍스트 파일은 연속적인 줄(line)들로 구성된다. **각 줄은 끝을 알리는 문자인 줄바꿈 문자(₩n)로 종료된다.**

이진 파일(binary file)

- 사람이 읽을 수 없는 컴퓨터가 읽을 수 있는 파일.
- 파일에 문자들이 아닌 이진 데이터가 저장되어 있다. 텍스트 파일처럼 한 줄이라는 개념이 없기 때문에 줄의 끝을 표시할 필요가 없다.
- 이진 파일은 특정한 프로그램에 의해서만 판독이 가능하다.
- **이미지, 사운드, 동영상, 실행 파일 등이 이진 파일들이다.**



텍스트 파일과 이진 파일의 장단점

예를 들어 정수 **1234**를 텍스트 파일로 저장하면 숫자 하나 하나가 문자로 변환된다.

정수 **1234** ➡ **"1"** 00110001 **"2"** 001100010 **"3"** 001100011 **"4"** 0011000100

- 장점 : 아스키 코드로 저장되므로 다른 컴퓨터에서도 읽을 수 있다.
- 단점 : 텍스트 파일에서 숫자 데이터를 읽으려면 먼저 문자를 읽고 **int()**로 변환해야 하므로 시간이 많이 걸리며 비효율적이다.

예를 들어 정수 **1234**를 이진 파일로 저장하면 문자열로 변환하지 않고 이진수 형태로 저장된다.

정수 **1234** ➡ 00000100 11010010

- 장점 : 이진 파일은 변환 과정이 없이 바로 숫자로 읽을 수 있으며 저장 공간도 적게 차지한다.
- 단점 : 인간이 파일 내용을 확인하기 어렵다. 그리고 이진 파일은 정수나 실수를 표현하는 방식이 컴퓨터 시스템마다 다를 수 있으므로 이식성이 떨어진다.

이식성이 중요하다면 텍스트 형식의 파일 사용.
파일이 크고 속도가 중요하다면 이진 파일 사용.

텍스트 파일 읽고 쓰기

- **파일 열기 : open("파일명" , "모드")**
  파일 읽기: 파일객체 = open("파일명" , "r")
  파일 쓰기: 파일객체 = open("파일명" , "w")

- **파일 닫기 : 파일객체.close()**
  더 이상 사용하지 않거나 프로그램을 종료할 때 열었던 파일은 닫아야 한다.
  닫지 않으면 해당 파일이 시스템 리소스를 차지하고 있으므로 운영체제 입장에서 보면 자원을 효율적으로 관리할 수 없게 된다

with 문

- 파일은 open()했으면 반드시 close()를 호출해서 닫아야 한다.
- with문을 사용하면 with문이 끝날 때 자동으로 close()가 호출된다.

사용법

```
with open(파일명, 모드) as 파일객체:
        처리할 명령문
```

```
f = open("foo.txt", "w")
f.write("You need python!")
f.close()
```

```
with open("foo.txt", "w") as f:
    f.write("You need python!")
```

텍스트 파일 읽고 쓰기(파일 모드)

- 파일 모드로 **r, w, a** 가 있다.

분류	종류	의미	설명	파일이 없을 때	파일이 있을 때
입력	r	read	읽기	오류	읽기
출력	w	write	쓰기	새로 생성	새로 생성
	a	append	추가	새로 생성	추가
	x	exclusive	배타적 추가	새로 생성	오류

종류	의미	설명
t	text	텍스트 파일
b	binary	바이너리 파일(텍스트 파일 외의 모든 파일)

파일 읽기/쓰기 관련 함수

텍스트 파일 읽기 함수

- **read()** : **파일 전체 또는 지정한 size만큼 읽기**

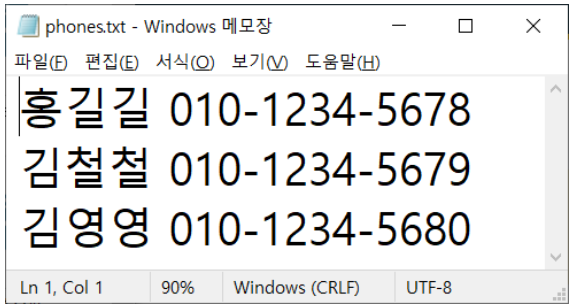
  용량이 큰 파일의 전체를 읽을 경우 메모리 공간에 문제가 발생할 수 있으므로 size로 크기를 지정하고 반복적으로 읽도록 구현하는 것이 좋다.

요소	텍스트 모드	바이너리 모드
반환값	읽어 들인 문자열	읽어 들인 바이트열
매개변수 size	읽어 들인 최대 문자의 개수	읽어 들일 최대 바이트 수
매개변수 size 생략	파일 전체 읽음	
파일의 끝에 도달	빈 문자열(" ")반환	

- **readline() : 한 줄씩 읽기**. **반복문이 필요함**.
- **readlines()** : **전체 라인을 모두 읽어 각 라인 단위로 리스트에 저장한다**.

텍스트 파일 읽기

1. 메모장을 열고 아래 내용을 작성하고 **phones.txt** 로 저장한다.



메모장은 utf-8로 기본 인코딩 된다.

2. 아래 코드를 입력하고 **phones.txt**를 읽는다.

```
with open("phones.txt","r",encoding="utf-8") as file:
    content=file.read()
    print(content)
```

인코딩을 utf-8로 설정하지 않으면 오류가 발생한다.

```
홍길길 010-1234-5678
김철철 010-1234-5679
김영영 010-1234-5680
```

```
file=open("phones.txt","r",encoding="utf-8")
content=file.read()
print(content)
file.close()
```

텍스트 파일 읽기 read()

- **read( )로 파일 전체 읽기**

```
with open("phones.txt","r",encoding="utf-8") as file:
    content=file.read()
    print(content)
```

- **read(size=크기)로 지정한 size만큼 읽기**

```
with open("phones.txt","r",encoding="utf-8") as file:
    while True:
        content=file.read(5)
        if not content:
            break
        print(content, end='')
```

텍스트 파일 읽기 readline()

- **readline( )로 한 줄씩 읽기**

```
with open("phones.txt","r",encoding="utf-8") as file:
    while True:
        content=file.readline()
        if not content:
            break
        print(content.strip())
```

텍스트 파일의 줄 끝에는
줄바꿈이 추가되어 있다.
strip() 는 좌우 공백 제거한다.

for 문을 사용하면 readline()함수가 필요 없다.

```
with open("wordlist.txt","r",encoding="utf-8") as file:
    for line in file:
        print(line.strip())
```

```
홍길길 010-1234-5678
김철철 010-1234-5679
김영영 010-1234-5680
```

텍스트 파일 읽기 readlines()

- **readlines( )**로 파일을 통째 읽어서 각 라인을 리스트로 저장해서 표시하기

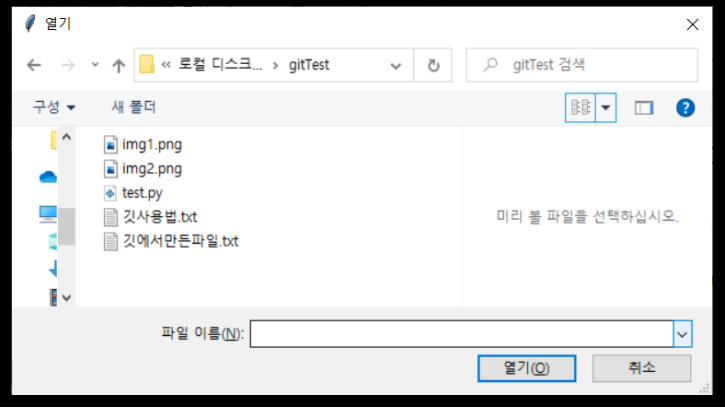
```
with open("phones.txt","r",encoding="utf-8") as file:
    line=file.readlines()
    print(line)
```

```
["홍길길 010-1234-5678₩n", "김철철 010-1234-5679₩n", "김영영 010-1234-5680"]
```

파일 읽기(파일 대화 상자 사용)

- 파일을 열고 쓸때 **tkinter**의 파일 열기 대화 상자를 활용할 수 있다.

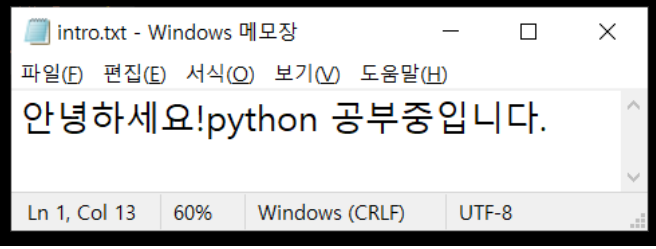
```
from tkinter.filedialog import askopenfilename

read_file=askopenfilename()
if read_file:
    infile=open(read_file,"r",encoding="utf-8")

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

infile.close()
```



텍스트 파일 쓰기 함수

- **write()** : 파일객체.write(문자열)

```
outfile=open("intro.txt","w",encoding="utf-8")
outfile.write("안녕하세요!")
outfile.write("python을 공부중입니다.")
outfile.close()
```



- **writelines()** : 파일객체.writelines(문자열 or 리스트)

writelines( )로 리스트를 저장할 경우
요소 값이 str(문자열)인 것만 가능함.

```
outfile=open("test.txt","w",encoding="utf-8")
li=["안녕하세요\n","감사합니다\n","thank you","see you"]
outfile.writelines(li)
```

write( ), writelines()함수는 줄 바꿈하지 않는다.
줄 바꿈을 원하면 ₩n을 직접 추가해야 한다.

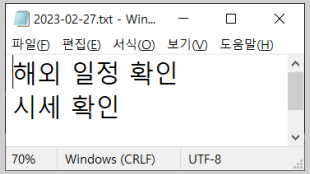
파일 입출력 응용

응용 - 오늘의 일정을 입력받아 오늘 날짜 이름의 파일로 저장

Python 2

1. 저장할 파일명인 오늘 날짜 파일명을 만든다.
2. 파일을 추가 모드로 연다.
3. 무한 반복문을 사용해서 사용자로부터 입력을 받는다. 아무것도 입력하지 않았으면 프로그램을 종료한다. 그렇지 않으면 입력된 행을 파일에 쓴다.
4. 파일을 닫는다.

```
오늘의 일정을 입력하세요: 해외 일정 확인
오늘의 일정을 입력하세요: 시세 확인
오늘의 일정을 입력하세요:
```

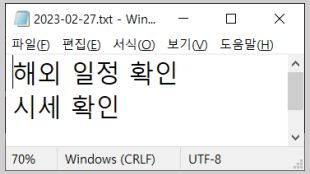


응용 - 오늘의 일정을 입력받아 오늘 날짜 이름의 파일로 저장

Python 2

```
import time

# 저장할 파일명 만들기(예 : 2025-01-01.txt)
f_name=time.strftime("%Y-%m-%d")+".txt"
# 파일을 추가 모드로 열기
f=open(f_name,"a")

# 아무것도 입력되지 않을 때까지 일정 입력받기
while True:
    schedule=input("오늘의 일정을 입력하세요: ")
    if not schedule:break
    f.write(schedule+"\n")
# 파일 닫기
f.close()
```

```
오늘의 일정을 입력하세요: 해외 일정 확인
오늘의 일정을 입력하세요: 시세 확인
오늘의 일정을 입력하세요:
```



응용 - 단어 퀴즈


```
# 단어 퀴즈
import random

with open("wordlist.txt","r",encoding="utf-8") as f:
    data=f.readlines()
    random.shuffle(data)

  while data:
    row=data.pop()
    quiz=row.strip().split("\t")
    print(quiz[0])
    answer=quiz[1].lower()
    your_answer=input("영어 입력: ")
    if answer==your_answer:
      print("O")
    else:
      print("X")
```


```
은하수
영어 문장 입력: galaxy
O
```

이진 파일 다루기

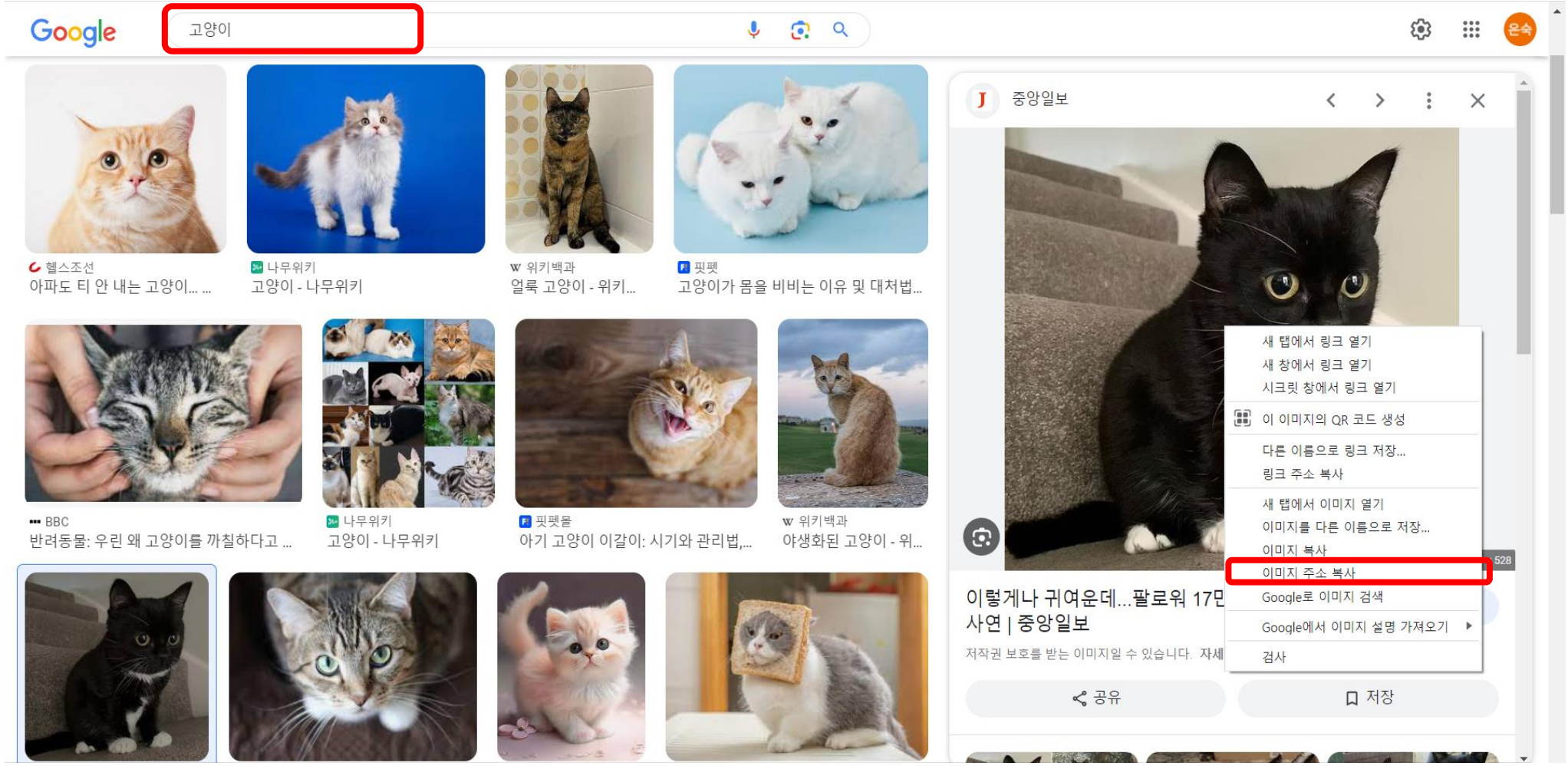
이진 파일 읽고 쓰기(열기와 닫기)

- **파일 열기 : open("파일명" , "모드")**
  파일 읽기: 파일객체 = open("파일명" , "**rb**")
  파일 쓰기: 파일객체 = open("파일명" , "**wb**")

```
source_file=open("pic2.gif","rb")           # pic2.gif- 38KB(38*1024)
copy_file=open("pic2_copy.gif","wb")
buffer=source_file.read(10240)              # 10KB 읽기
copy_file.write(buffer)
print("복사 완료.")
```

지정한 주소에서 이미지 받아 오기

1. 구글에서 검색어 창에 "고양이"입력해서 검색된 이미지 중 하나를 클릭한다.
2. 마우스 오른쪽 버튼을 클릭하고 "이미지 주소 복사"를 선택한다.



지정한 주소에서 이미지 받아 오기

3. 아래 소스를 작성하고 실행하면 앞서 선택한 이미지를 파일로 저장할 수 있다. request.get()에 이미지 주소 url을 지정한다.

```
import requests

img_addr="https://pds.joongang.co.kr/news/component/htmlphoto_mmdata/202306/25/488f9638-800c-4bac-ad65-82877fbff79b.jpg"

response=requests.get(img_addr)
with open('cat.png','wb') as f:
    f.write(response.content)
```

requests 라이브러리의 **get()** 메서드로 받은 응답(Response) 객체는 **text** 속성과 **content** 속성을 제공한다.
text는 응답으로 받은 데이터가 문자열일 때 사용한다.
content는 응답으로 받은 데이터를 이진 데이터(이미지, 동영상, 사운드 등)인 경우 사용한다.

weatherapi.com 날씨 아이콘을 이미지 파일로 저장

Python 2

```
import requests

city=input("도시명:")
res=requests.get("http://api.weatherapi.com/v1/current.json?key=API key&q="+city)
# print(res.text)

# 문자열인 데이터를 파이썬의 딕셔너리로 변경
data=res.json()    # json() : str->dict

# 날씨 아이콘 추출 및 이미지 저장
icon_addr="https:"+data["current"]["condition"]["icon"]
img=requests.get(icon_addr)   # 이미지 주소로 데이터 요청
with open("wicon.png", "wb") as fp:
    fp.write(img.content)
```