

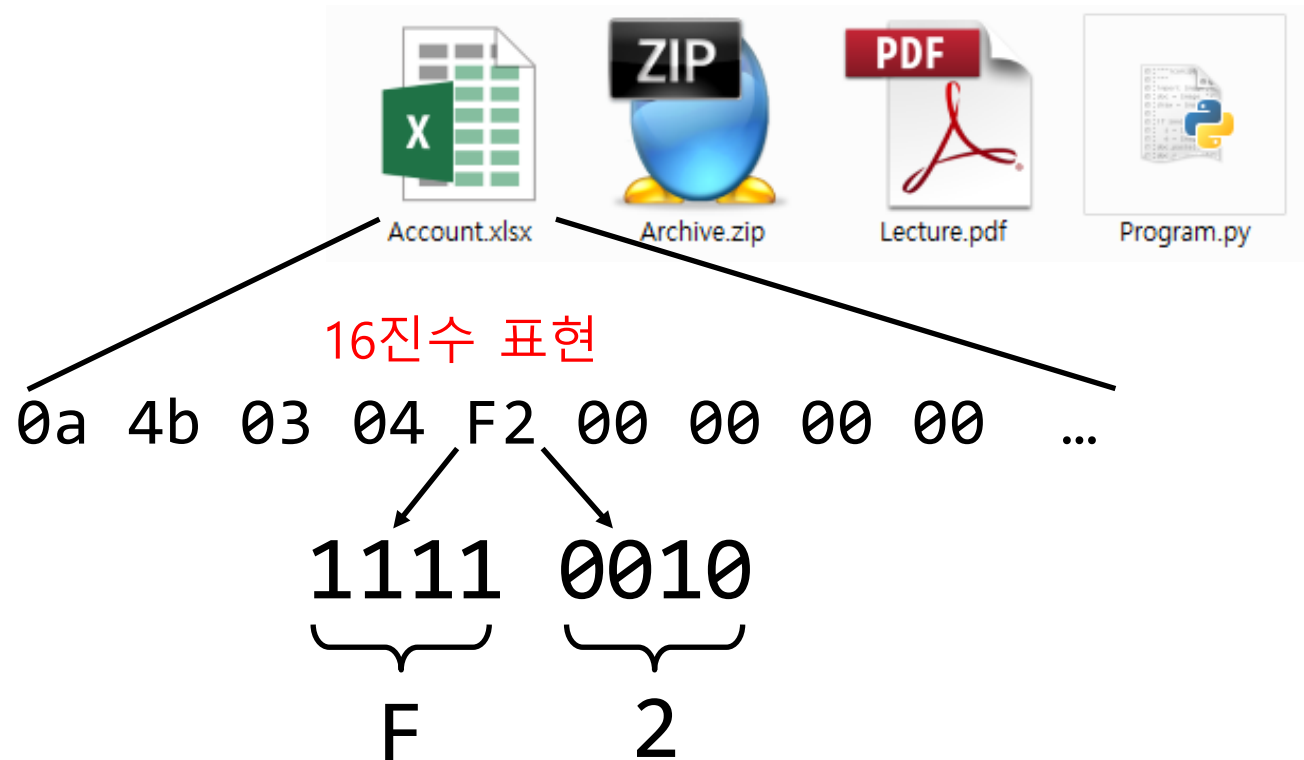
# 오늘의 강의 목표

---

- File에 대한 이해
- Python의 File I/O 방법에 대한 이해

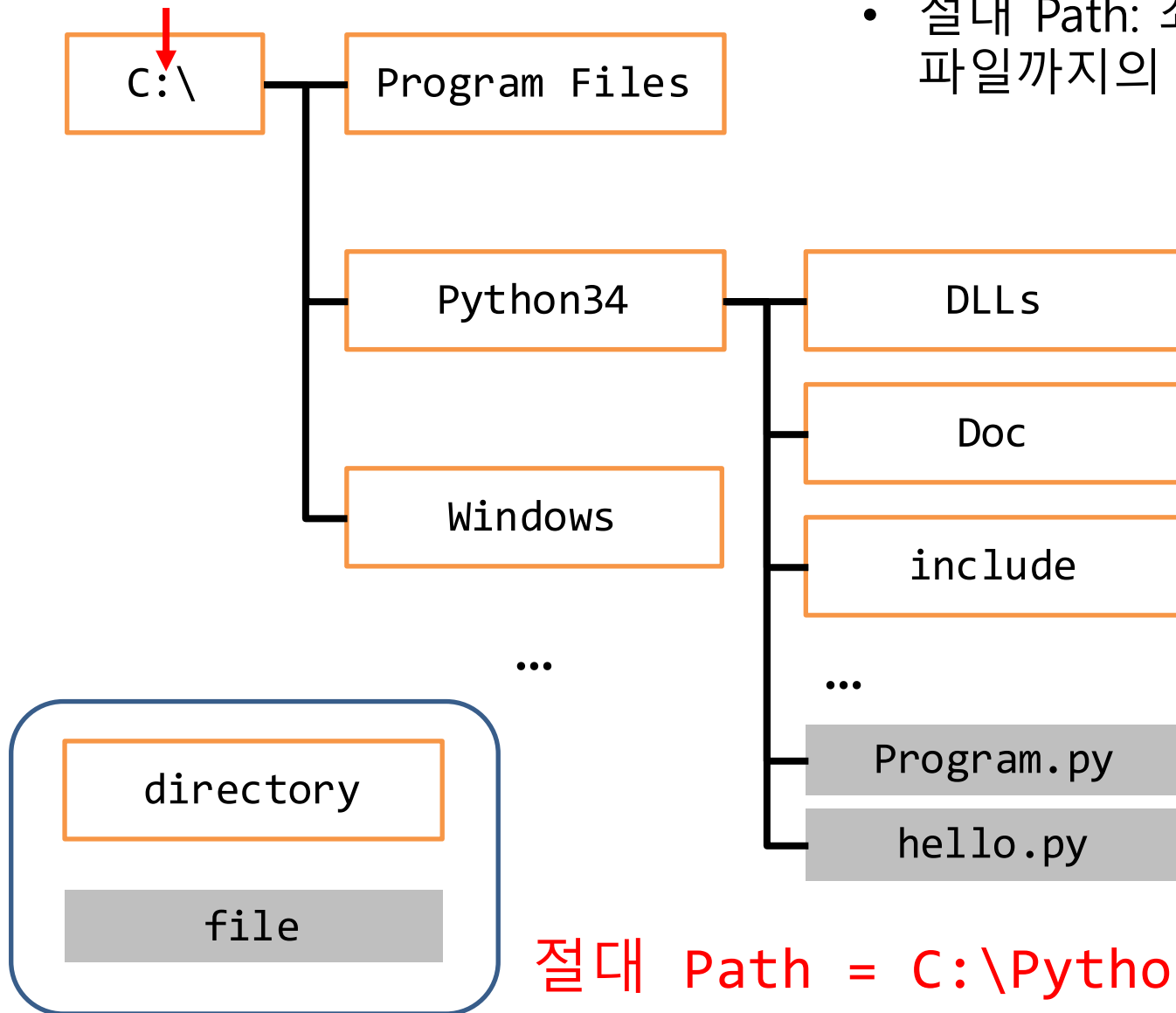
# Files

- File is just a sequence of bytes



# Directory and File Path (파일 경로)

최상위 Directory



- 절대 Path: 최상위 디렉토리부터 파일까지의 모든 디렉토리를 나열

절대 Path = C:\Python34\hello.py

# 절대 Path vs 상대 Path

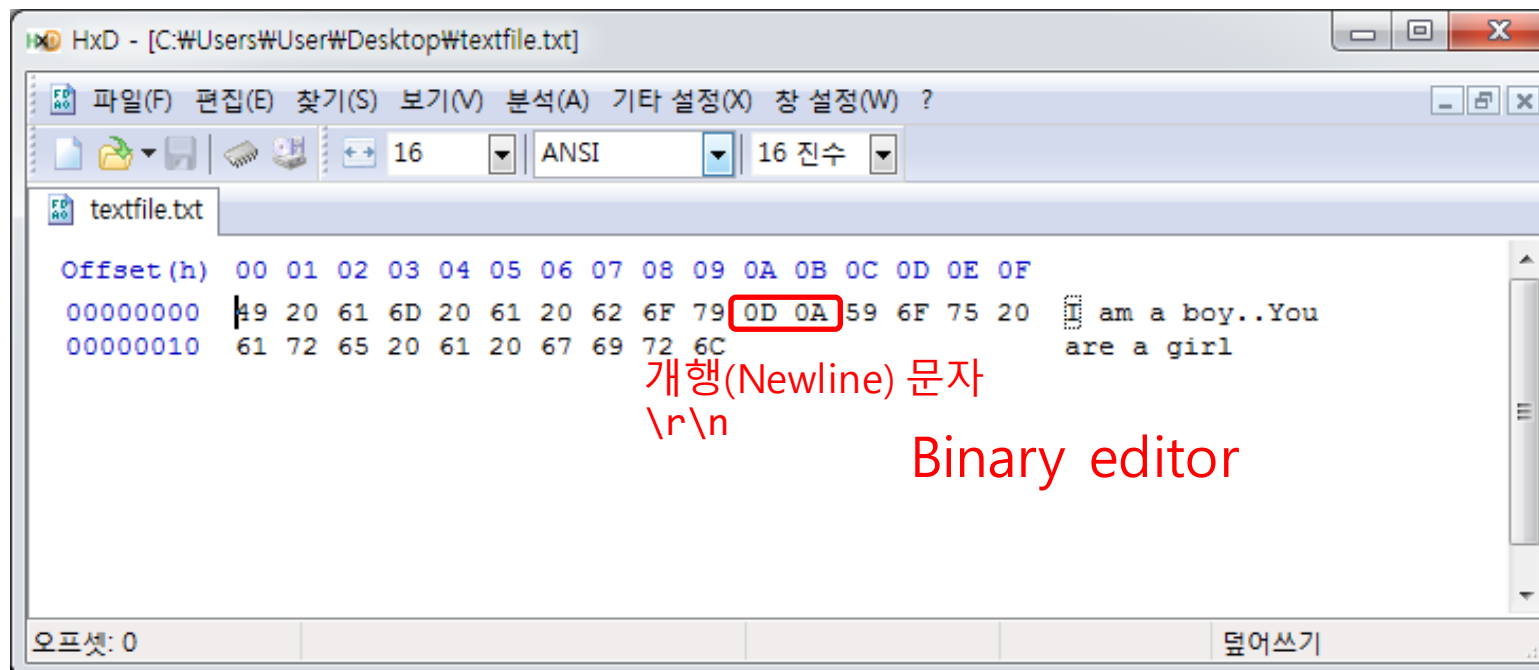
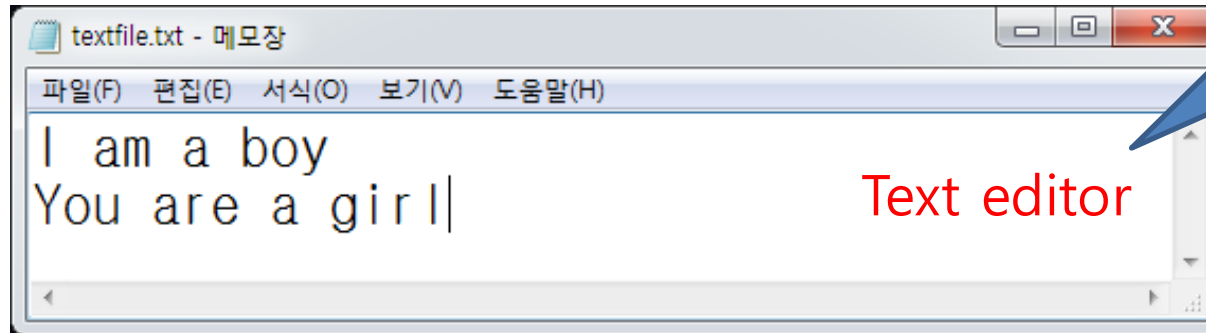
---

- 절대 Path (절대 경로)
    - 최 상위 디렉토리부터 파일까지의 모든 디렉토리를 나열
    - ex) C:\Python34\hello.py
  - 상대 Path (상대 경로)
    - 현재 위치를 기준으로 상대적인 위치를 명시  
Current Working Directory (CWD)
    - CWD가 C:\Python34라 할 때
      - ex) hello.py → C:\Python34\hello.py
      - ex) Lib\io.py → C:\Python34\Lib\io.py
      - ex) ../log.txt → C:\log.txt
- .. : 상위 디렉토리를 의미

# Text Files

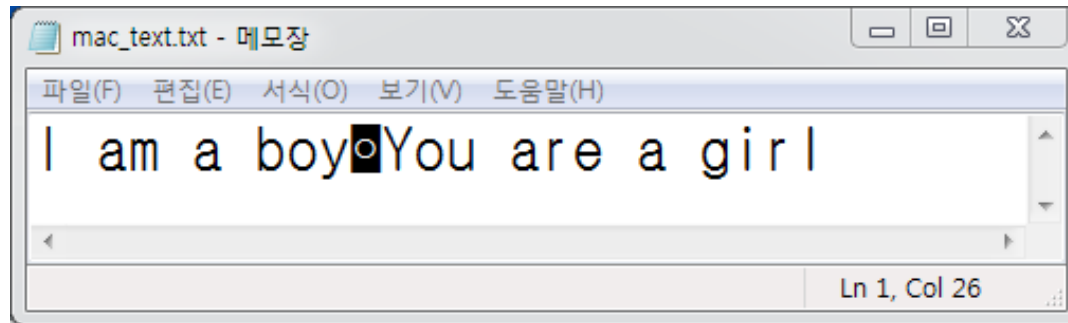
- Sequence of valid characters

매 Byte를 ASCII  
Code로 해석하여  
보여줌

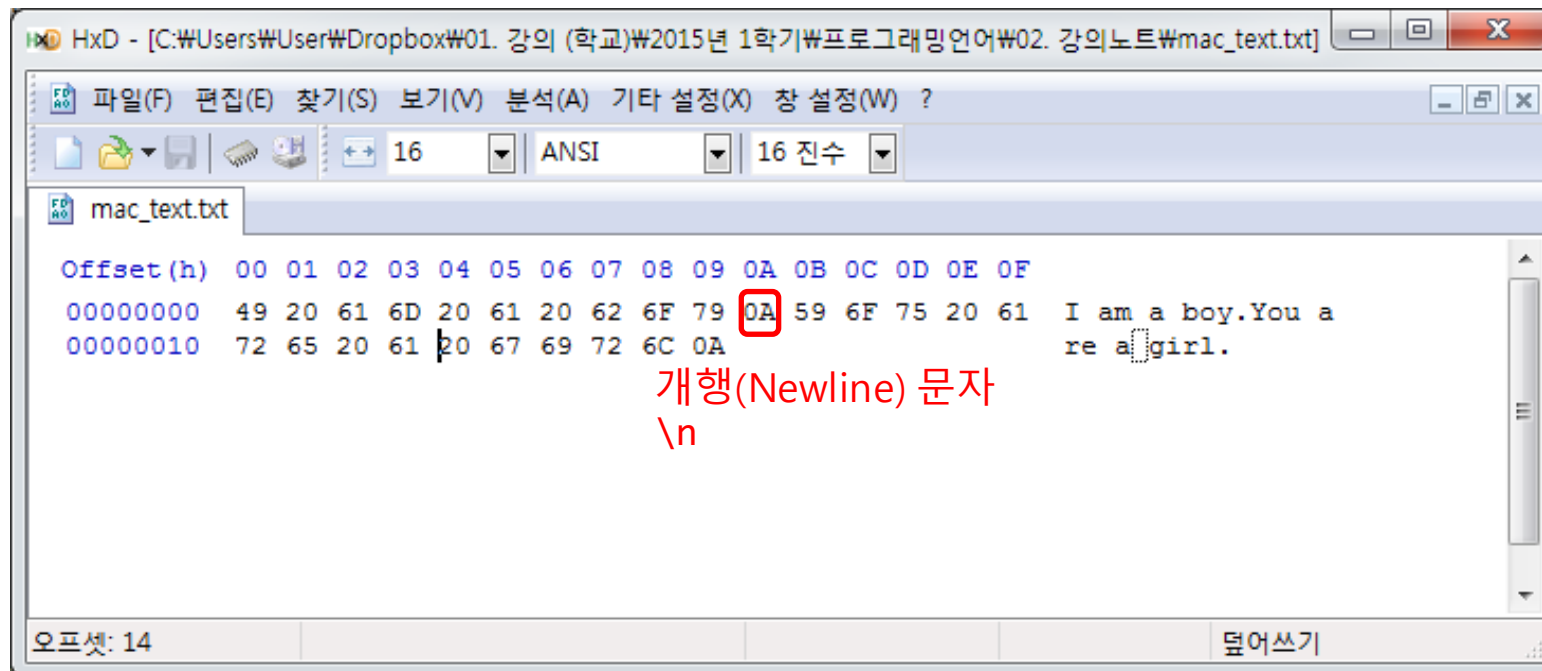


# 주의: 개행 문자

- OS에 따라 개행 문자가 상이할 수 있음



Mac OS에서 작성한 파일



# 주의: 개행 문자

---

- 윈도우의 개행 문자 : `\r\n` (2 bytes)
- 다른 OS의 개행 문자 : `\n` (1 byte)
- Python의 정책
  - 윈도우의 경우
    - `\r\n`을 `\n`으로 변환하여 읽어줌
    - `\n`을 쓸 때는 자동으로 `\r\n`으로 변환
  - 다른 OS의 경우
    - 그대로 읽고 씀

# open

---

- File을 읽고 쓸 수 있도록 open

```
open(file, mode)
```

```
>>> f = open("textfile.txt", "r")
```

```
>>> f = open("C:\Python34\textfile.txt", "r")
```

Character	Meaning
'r'	open for reading (default)
'w'	open for writing, truncating the file first
'x'	open for exclusive creation, failing if the file already exists
'a'	open for writing, appending to the end of the file if it exists
'b'	binary mode
't'	text mode (default)
'+'	open a disk file for updating (reading and writing)
'U'	<i>universal newlines</i> mode (deprecated)



# open

---

- 존재하지 않는 File의 경우 에러 발생

```
>>> f = open("thisfile.txt", "r")
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    f = open("thisfile.txt", "r")
FileNotFoundError: [Errno 2] No such file or
directory: 'thisfile.txt'
```

# readlines

---

- 행 단위로 읽어서 list로 리턴

```
file.readlines()
```

```
>>> f = open("textfile.txt", "r")
>>> lines = f.readlines()
>>> lines
['I am a boy\n', 'You are a girl']
```

# close

---

- File 닫기

```
file.close()
```

```
>>> f.close()
```

open → read & write → close

# readline

---

- Line 단위로 읽기

```
file.readline()
```

```
>>> f = open("textfile.txt", "r")
>>> f.readline()
'I am a boy\n'
>>> f.readline()
'You are a girl'
>>> f.readline()
''
```

↑  
Empty string means there is no more line to read

# for

---

- for를 이용한 file에 대한 line 단위 접근

```
>>> f = open("textfile.txt", "r")
>>> for line in f:
    print "[" + line + "]"
```

```
[I am a boy
]
[You are a girl]
```

# read

---

- Byte 단위 읽기

```
file.read(n)
```

```
>>> f = open("textfile.txt", "r")
>>> f.read()      # 전체 읽음
'I am a boy\nYou are a girl'
```

```
>>> f = open("textfile.txt", "r")
>>> f.read(7)     # 7 bytes 읽고 위치 전진
'I am a '
>>> f.read(3)     # 3 bytes 읽음 위치 전진
'boy'
```

# tell

---

- File의 현재 위치 리턴

```
file.tell()
```

```
>>> f = open("textfile.txt", "r")
>>> f.read(7)
'I am a '
>>> f.tell()
7
>>> f.read(3)
'boy'
>>> f.tell()
10
>>> f.read(1)
'\n'    # Newline도 하나의 문자로 취급
```

# seek

---

- File의 현재 위치 조정

`file.seek(offset, whence)`

0: 절대위치  
1: 현재위치 기준  
2: 파일의 끝 기준

```
>>> f = open("textfile.txt", "r")
>>> f.read(7)
'I am a '
>>> f.seek(2, 0)
2
>>> f.read(2)
'am'
```



# write

---

- File 쓰기

`file.write()`

없으면 새로 만듦  
있으면 기존 내용 삭제

```
>>> f = open("outputfile.txt", "w")
>>> name = "Kim"
>>> score = 98
>>> f.write("%s's score is %d\n" % (name, score))
18
>>> f.close()
```

# Command-line Arguments

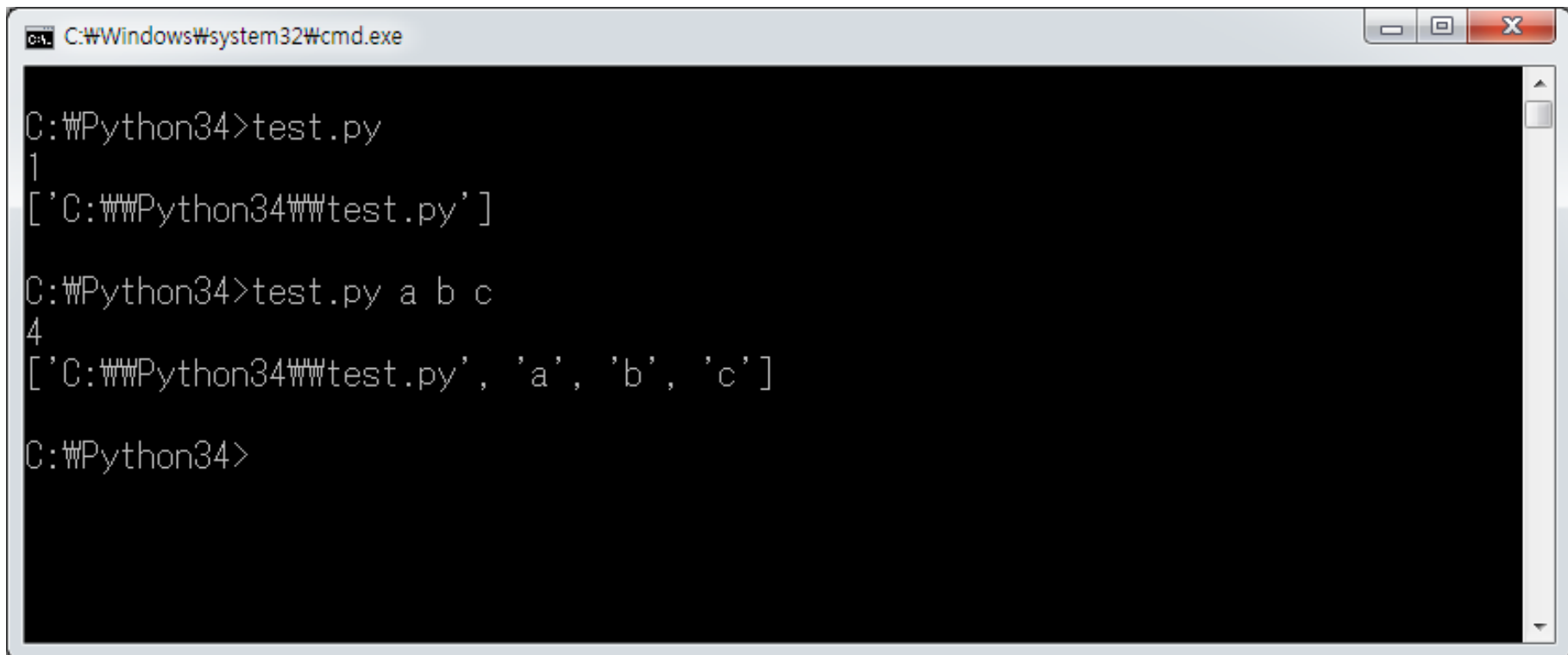
---

- `sys.argv`를 이용하면 명령행 인자를 받을 수 있음

```
import sys
```

```
print(len(sys.argv))
```

```
print(sys.argv)
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The prompt is at "C:\Python34>". The first command executed is "test.py", which outputs "1" and "['C:\Python34\test.py']". The second command is "test.py a b c", which outputs "4" and "['C:\Python34\test.py', 'a', 'b', 'c']".

```
C:\Windows\system32\cmd.exe
C:\Python34>test.py
1
['C:\Python34\test.py']
C:\Python34>test.py a b c
4
['C:\Python34\test.py', 'a', 'b', 'c']
C:\Python34>
```

# Text File Copy

---

- a를 b로 복사하는 copy.py 파일 작성

```
import sys
```

```
f1 = open(sys.argv[1], "r")
```

```
f2 = open(sys.argv[2], "w")
```

```
f2.write(f1.read())
```

```
f1.close()
```

```
f2.close()
```

one-shot copy

# Text File Copy

---

- a를 b로 복사하는 copy.py 파일 작성

```
import sys
```

```
f1 = open(sys.argv[1], "r")
```

```
f2 = open(sys.argv[2], "w")
```

```
for line in f1:
```

```
    f2.write(line)
```

```
f1.close()
```

```
f2.close()
```

Line-by-line copy

# Text File Copy

---

- a를 b로 복사하는 copy.py 파일 작성

readline() 이용

# Practice: Upper & Lower

---

- a를 b로 복사하되 모든 소문자는 대문자로, 모든 대문자는 소문자로 변경

# Advanced Topics

---

- File open modes
- Pickle
- Directory handling

# Questions

---

