

Chapter 9

파일 읽기/쓰기

임경태



Preview



학습목표

- 파일을 읽는 방법을 살펴봅니다.
- 읽은 파일의 내용을 처리하는 방법을 익힙니다.
- 파일을 쓰는 방법을 살펴봅니다.
- 파일을 읽어서 동일한 파일로 쓰는 방법을 학습합니다.

Section 01

파일을 사용하는 이유

파일은 왜 필요한가?



■ 훈민정음을 매번 입력해야하는 경우

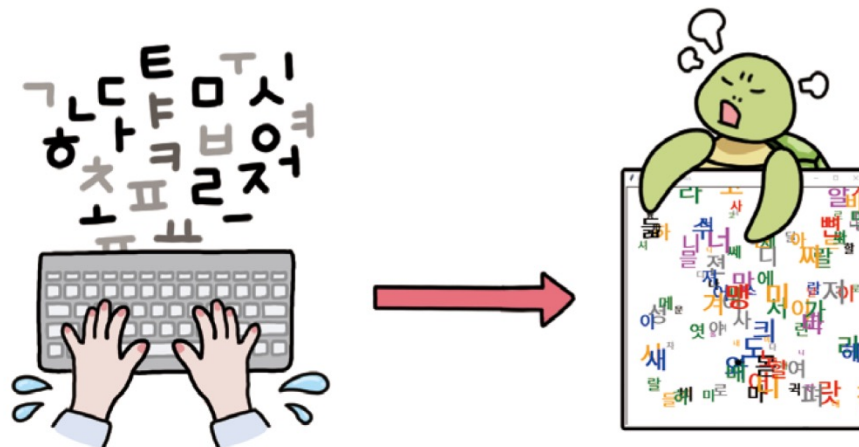


그림 9-1 키보드를 사용한 훈민정음 입력 작업의 반복

■ 훈민정음을 파일로 저장해서 읽는 경우

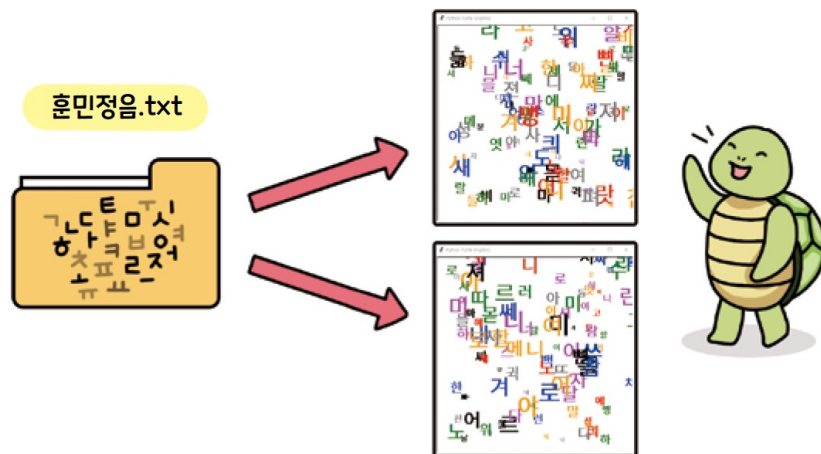


그림 9-2 파일에서 훈민정음을 읽어서 화면에 출력

파일은 왜 필요한가?



■ 실행 결과를 파일로 저장하지 않을 경우

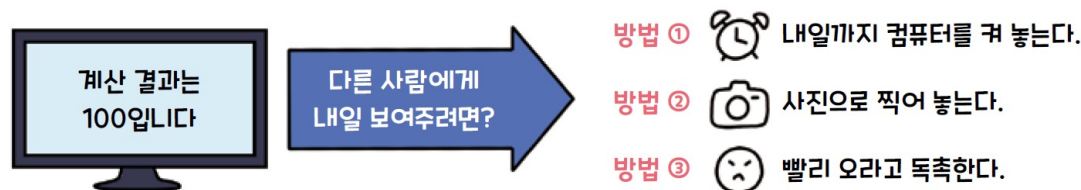


그림 9-3 화면의 출력 결과를 다른 사람에게 보여줄 때

■ 실행 결과를 파일로 저장할 경우

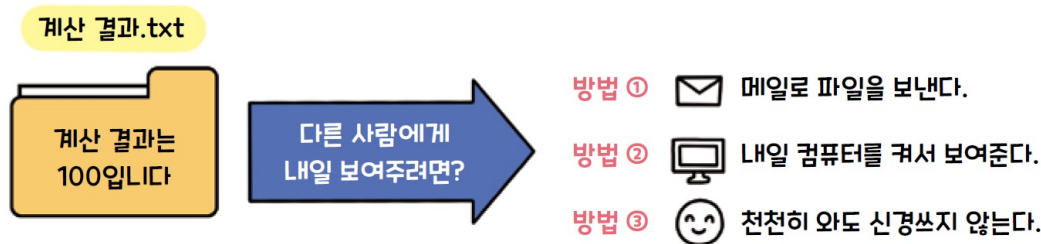


그림 9-4 파일에 저장된 출력을 다른 사람에게 보여줄 때

파일은 왜 필요한가?



확인문제

다음 빈칸에 들어갈 단어를 (모니터, 파일) 중에 골라 채우시오.

에 출력하면 한 번만 보고 없어지지만, 에 출력하면 계속 반복해서 내용을 확인할 수 있다.

정답

Click!

Section 02

파일 읽기

파일 읽기의 3단계



■ 파일 읽기 3단계

- ① 파일 열기
- ② 파일 읽기
- ③ 파일 닫기



1단계
파일 열기



2단계
파일 읽기



3단계
파일 닫기

그림 9-5 파일 읽기 3단계



■ 파일 열기

- 파일을 열기 위해서는 `open()` 함수에서 파일명을 지정하고, 읽기(Read)를 의미하는 "r"로 설정함
- 모드(mode) : `open()` 함수의 마지막 매개변수
 - 파일을 열 때 어떤 용도로 열지 결정함
 - 파일 읽기용은 "r"을 씀

```
변수명 = open("파일경로/파일이름", "r")
```



■ 파일 읽기

- 파일에서 데이터를 읽어올 수 있는 상태
 - 파일에서 데이터를 읽는 세부적인 방법은 잠시 후 예제를 통해 살펴봄

■ 파일 닫기

- 파일과 관련된 모든 작업이 끝나면 파일을 정상적으로 닫아줘야 함
- 파일을 닫기 위해 사용하는 변수는 `open()` 함수로 열었던 변수명임

```
변수명.close();
```

파일 읽기의 3단계



확인문제

다음 파일 읽기의 3단계를 순서대로 나열하시오.

- (A) 파일 읽기
- (B) 파일 닫기
- (C) 파일 열기

정답

Click!

readline() 함수



- 파일에 담아둔 데이터를 읽기 위한 함수
 - readline() , readlines()



그림 9-6 파일을 읽어오는 함수



■ 한 행씩 읽기

- readline() : 파일의 내용을 한 행씩 읽어옴
- 파일에 100개의 행이 있다면, 100번을 반복해서 읽어야 함

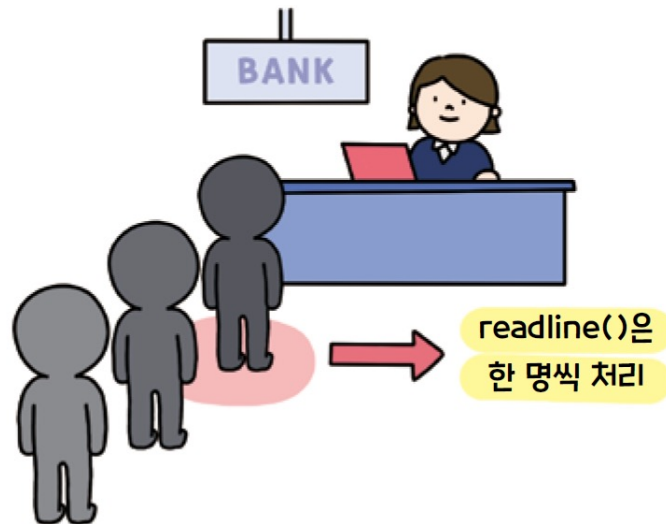


그림 9-7 한번에 한 행씩만 처리하는 readline() 함수



■ readline() 함수 예제

- ① 메모장을 실행하여 세 줄의 내용 채우기
 - 파일 이름: myData1.txt
 - 파일 경로: C:\FirstPython\Chapter09

난생처음 파이썬을 공부합니다.
코딩이 재미있어 졌어요.
이젠 코딩 전문가가 된 것 같아요. ^^



■ readline() 함수 예제

② 코드 작성하고 실행하기

[코드 9-1]

```
inFile = None # 입력 파일
inStr = ""     # 읽어온 문자열

inFile = open("C:/FirstPython/Chapter09/myData1.txt", "r",
encoding="UTF-8")

inStr = inFile.readline()
print(inStr, end='')

inStr = inFile.readline()
print(inStr, end='')

inStr = inFile.readline()
print(inStr, end='')

inFile.close()
```

[실행결과]

난생처음 파이썬을 공부합니다.
코딩이 재미있어 졌어요.
이젠 코딩 전문가가 된 것 같아요. ^^



■ 파일의 모든 행 읽기

- [코드 9-1]은 파일의 내용 중, 정확히 3개 행만 처리가 가능함
 - 따라서 파일에 1개 또는 2개 행만 있으면 오류가 발생함
- 파일에 4개 행 이상이 있다면 오류는 발생하지 않지만 3개 행까지만 읽음
- 파일에 있는 모든 행을 다 읽기 위해서 [코드 9-1] 수정하기

[코드 9-2]

```
inFile = None
inStr = ""

inFile = open("C:/FirstPython/Chapter09/myData1.txt", "r",
encoding="UTF-8")

while True :
    inStr = inFile.readline()
    if inStr == "" :
        break
    print(inStr, end='')

inFile.close()
```



확인문제

다음 중 잘못된 것을 모두 고르시오.

- ① readline() 함수는 파일에서 한 행씩 읽는 함수이다.
- ② readline() 함수는 파일을 쓸 때 사용하는 함수이다.
- ③ readline() 함수를 사용하면 파일에 몇 행의 문장이 있든지 전부 읽는다.
- ④ readline() 함수를 사용하여 파일의 내용을 모두 읽으려면 무한 반복을 사용한다.

정답

Click!



■ 파일의 내용 한꺼번에 읽기

- `readlines()` : 파일의 내용을 한꺼번에 읽어 리스트에 저장함

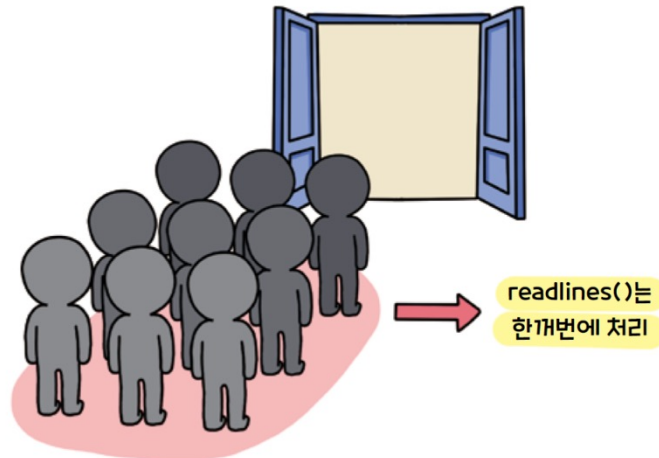


그림 9-8 여러 행을 한꺼번에 처리하는 `readlines()` 함수



■ readlines() 함수 예제

[코드 9-3]

```
inFile = None
inList = []

inFile = open("C:/FirstPython/Chapter09/myData1.txt", "r",
encoding="UTF-8")

inList = inFile.readlines()
print(inList)

inFile.close()
```

[실행결과]

['난생처음 파이썬을 공부합니다.\n',
'코딩이 재미있어 졌어요.\n', '이젠 코딩
전문가가 된 것 같아요. ^^']

출력 결과에서 각 문자열 뒤에 붙은 '\n'은 myData1.txt 파일을 작성할 때, 각 행의 끝에서 **Enter**를 눌러서 다음 줄로 넘어갔다는 의미입니다.



■ readlines() 함수 예제

- [코드 9-3]의 print() 함수를 수정해서 파일의 내용이 한 행씩 출력하기

[코드 9-4]

```
inFile = None
inList = []

inFile = open("C:/FirstPython/Chapter09/myData1.txt", "r",
encoding="UTF-8")

inList = inFile.readlines()
for inStr in inList :
    print(inList, end = "")

inFile.close()
```

[실행결과]

난생처음 파이썬을 공부합니다.
코딩이 재미있어 졌어요.
이젠 코딩 전문가가 된 것 같아요. ^^

파일의 내용을 모니터에 출력할 때, 몇 번째 행에 어떤 내용이 있는지 알아볼 수 있도록 제일 앞에 행 번호가 붙도록 출력해봅시다.

실행 결과

- 1 : 난생처음 파이썬을 공부합니다.
- 2 : 코딩이 재미있어 졌어요.
- 3 : 이젠 코딩 전문가가 된 것 같아요. ^^



1. 파일의 내용을 모니터에 출력할 때, 몇 번째 행에 어떤 내용이 있는지 알아볼 수 있도록 제일 앞에 행 번호가 붙도록 출력하기

```
inFile = None
inStr = ""
lineNum = 1

inFile = open("C:/FirstPython/Chapter09/myData1.txt", "r",
encoding="UTF-8")
```

2. 한 줄씩 읽어 들이며 한 줄을 출력할 때마다 행 번호를 1씩 증가시키기
 - 행 번호와 읽어온 문자열 출력하기

```
while True :
    inStr = inFile.readline()
    if inStr == "" :
        break
    print(lineNum, " : " , inStr, end='')
    lineNum += 1

inFile.close()
```

3. <Ctrl>+<S>를 눌러서 변경된 내용을 저장하고, <F5>를 눌러 실행 결과 확인하기

Section 03

파일 쓰기



■ 파일 쓰기

- 결과를 파일에 쓰게 되면 모니터에는 나오지 않고, 파일로 직접 저장됨
- `writelines()` : 파일에 결과를 씀

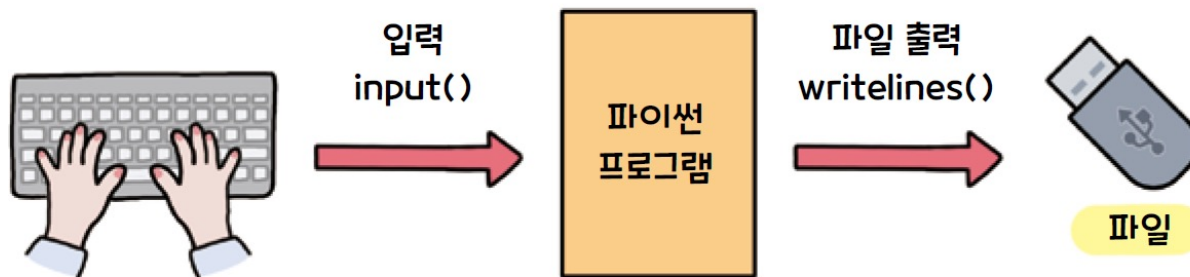


그림 9-9 키보드 입력과 파일 출력



■ 파일 열기

- 파일을 쓰기 위해서도 `open()` 함수에서 파일명을 지정하고, 파일 쓰기 모드를 지정
- 이때 파일 경로에 같은 이름의 파일이 있다면 기존 파일을 덮어씀
 - 같은 이름의 파일이 없다면 파일을 새로 생성함
- 쓰기용은 `open()` 함수의 매개변수로 "w"를 사용함

```
변수명 = open("파일경로/파일이름", "r")
```

■ 파일 쓰기

■ 파일 닫기

```
변수명.close();
```



확인문제

다음 파일 쓰기 3단계를 순서대로 나열하시오.

- (A) 파일 쓰기
- (B) 파일 닫기
- (C) 파일 열기

정답

Click!



■ writelines() 함수 예제

[코드 9-5]

```
outFile = None
outStr = ""

outFile=open("c:/FirstPython/Chapter09/myData2.txt", "w")

outStr = "안녕하세요?"
outFile.writelines(outStr+"\n")

outStr = "반갑습니다."
outFile.writelines(outStr+"\n")

outStr = "자주만나요. ^^"
outFile.writelines(outStr+"\n")

outFile.close()
print("--- myData2.txt 파일이 저장됨 ---")
```

[실행결과]

--- myData2.txt 파일이 저장됨 ---

사용자에게 입력받은 내용을 파일에 쓰기



■ 사용자가 키보드에서 입력한 내용을 한 행씩 파일에 쓰기

[코드 9-6]

```
outFile = None
outStr = ""

outFile = open("c:/FirstPython/Chapter09/myData3.txt", "w")

while True:
    outStr = input("내용 입력 ==> ")
    if outStr != "" :
        outFile.writelines(outStr+"\n")
    else :
        break

outFile.close()
print("--- myData3.txt 파일이 저장됨 ---")
```

[실행결과]

```
내용 입력 ==> 난생처음
내용 입력 ==> 파이썬을
내용 입력 ==> 열공 중입니다.
내용 입력 ==> 
--- myData3.txt 파일이 저장됨 ---
```

사용자 입력

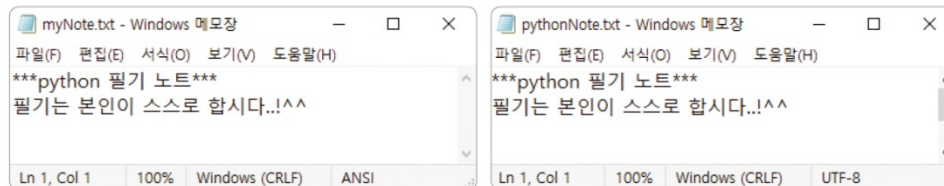
친구의 파이썬 수업의 필기 노트 파일을 복사하려고 합니다.

파일을 띄워놓고 한 줄씩 따라 쓰기에는 시간이 부족하니 친구의 필기 노트 파일과 동일한 파일을 복사해서 만드는 프로그램을 작성해 봅시다. 윈도우의 파일 탐색기에서 파일을 선택한 후, <Ctrl>+<C>로 복사하고 <Ctrl>+<V>로 붙여넣기하는 효과와 동일합니다



실행 결과

--- pythonNote.txt가 myNote.txt로 복사되었음 ---



1. lab09-02.py 파일을 만들고, 읽기 모드로 복사할 파일을 열고 쓰기 모드로는 붙여넣을 새로운 파일 열기

```
inFile, outFile = None, None
inStr = ""

inFile=open("C:/FirstPython/Chapter09/pythonNote.txt", "r",
encoding="UTF-8")
outFile=open("C:/FirstPython/Chapter09/myNote.txt", "w")
```

2. inFile의 내용을 한꺼번에 읽은 후에, 읽어온 리스트의 내용을 한 줄씩 outFile에 쓰기

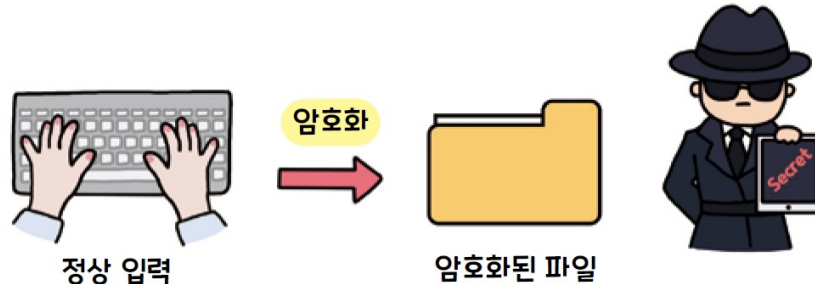
```
inList = inFile.readlines()
for inStr in inList :
    outFile.writelines(inStr)
```

3. 붙여넣는 반복이 끝나면 파일 닫기

```
inFile.close()
outFile.close()
print("--- PythonNote.txt가 myNote.txt로 복사되었음 ---")
```

4. 파일을 생성한 경로를 열어보면 myNote.txt 파일을 볼 수 있음

중요한 내용의 보안이 필요한 파일을 메일로 전달할 때, 그냥 전달하면 누출의 위험이 있습니다. 그래서 입력한 내용을 파일로 암호화해서 저장하는 방법을 살펴봅시다



실행 결과

```
스파이에게 전달할 메시지 ==>이번 임무는  
스파이에게 전달할 메시지 ==>파이썬 프로그래밍 수업에서  
스파이에게 전달할 메시지 ==>100점 맞기입니다.  
스파이에게 전달할 메시지 ==>그럼, 다음에 또...  
스파이에게 전달할 메시지 ==>Enter  
--- secure.txt 암호화 완료 ---
```

사용자 입력

- 글자 암호화 하는 방법

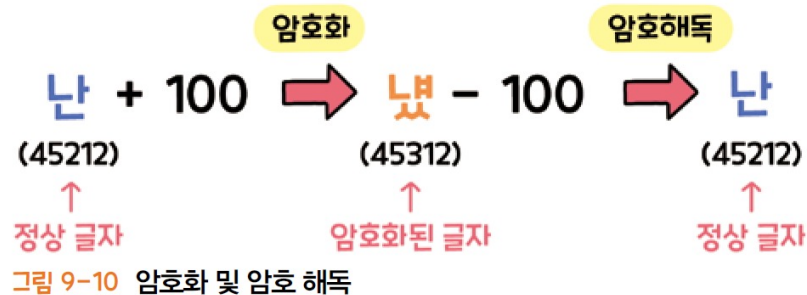
- ord(문자) : 문자의 고유 숫자를 알려줌
- chr(숫자) : 고유 숫자에 해당하는 문자를 알려줌

```
>>> ord('난')
45212
>>> chr(45212)
'난'
```

- 암호화 : '난'의 고유 숫자인 45212에 적당한 값 더하기
- 해독하기 : '넛'의 고유 숫자인 45312에서 100을 빼기

```
>>> num = ord('난')
>>> chr(num+100)
'넛'
```

```
>>> num = ord('넛')
>>> chr(num-100)
'난'
```



1. lab09-03.py 파일을 만들고, 암호화해서 저장할 파일을 준비하기

- 입력할 문자열 inStr과 암호를 저장할 문자열 secure를 준비하기

```
secureFile = None
inStr, secure = "", ""

secureFile=open("C:/FirstPython/secure.txt", "w", encoding="UTF-8")
```

2. 사용자가 아무것도 입력하지 않고, <Enter>를 누를 때까지 무한 반복하기

```
while True :
    inStr = input('스파이에게 전달할 메시지 ==>')
    if inStr == "" :
        break
```

3. 정상 문자열 inStr의 숫자에 100씩 더해 문자열 secure에 저장하기

```
for ch in inStr :
    num = ord(ch)
    num += 100
    secure += chr(num)
```

4. 암호화한 문자열을 파일에 쓰고 닫기

```
secureFile.writelines(secure)
secureFile.close()
print('--- secure.txt 암호화 완료 ---')
```

4. <Ctrl> + <S>를 눌러서 변경된 내용을 저장하고, <F5>를 눌러서 실행하고 스파이에게 전달할 메시지를 입력하기

실행 결과

```
스파이에게 전달할 메시지 ==>이번 임무는
스파이에게 전달할 메시지 ==>파이썬 프로그래밍 수업에서
스파이에게 전달할 메시지 ==>100점 맞기입니다.
스파이에게 전달할 메시지 ==>그럼, 다음에 또...
스파이에게 전달할 메시지 ==>Enter
--- secure.txt 암호화 완료 ---
```

} 사용자 입력

5. 저장된 secure.txt를 열어보면 암호화된 것을 확인할 수 있음

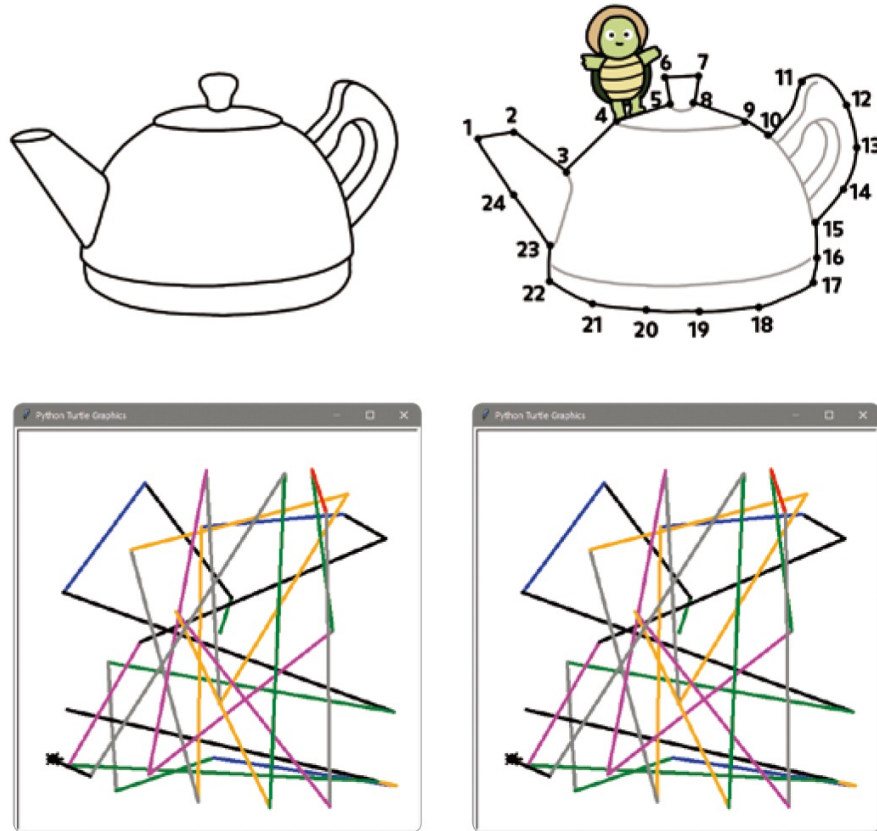
```
잠벳   잔뽀뽀잠벳   잔뽀뽀뽀잠벳   함뽀뽀뽀뽀뽀잠벳   잔뽀뽀뽀잠벳   함뽀뽀뽀뽀뽀뽀
존   뽀뽀잔뽀뽀   잠벳   잔뽀뽀뽀잠벳   함뽀뽀뽀뽀뽀   존   뽀뽀잔뽀뽀   잠론   뽀잔뽀
뽀
```

[실전 예제] 갔던 길을 복습하는 거북이

실전 예제 갔던 길을 복습하는 거북이

[문제]

- 거북이가 화면 안의 랜덤한 위치로 30회 이동하고, 화면을 깨끗하게 지운 후에, 바로 앞에서 이동한 경로와 동일하게 이동시켜 보자.
- 이동한 경로를 파일로 기록하여 그대로 거북이가 움직임



실전 예제 갔던 길을 복습하는 거북이

[해결]

```
import turtle
import random
import time

colorList = ["red", "green", "blue",
             "black", "magenta", "orange", "gray"]
turtle.shape('turtle')
turtle.setup(550, 550)
turtle.screensize(500, 500)
turtle.pensize(5)

turtleFile =
open("C:/FirstPython/Chapter09/turtleTrace
.txt", "w")

for _ in range(30) :
    x = random.randint(-250, 250)
    y = random.randint(-250, 250)
    color = random.choice(colorList)
    turtle.pencolor(color)
    turtle.goto(x,y)
    outStr = str(x) + " " + str(y) + " " +
color + "\n"
    turtleFile.writelines(outStr)
```

```
turtleFile.close( )
turtle.reset( )

time.sleep(5)
turtle.pensize(5)
turtleFile =
open("C:/FirstPython/Chapter09/turtleTrace
.txt", "r")

while True :
    inStr = turtleFile.readline( )
    if inStr == "" :
        break
    x, y, color = inStr.split( )
    turtle.pencolor(color)
    turtle.goto(int(x),int(y))

turtleFile.close( )
turtle.done( )
```

감사합니다 :)