

UNIT 27

파일 사용하기

27.1 파일에 문자열 쓰기, 읽기

» 파일에 문자열 쓰기

- 파일에 문자열을 쓸 때는 open 함수로 파일을 열어서 파일 객체(file object)를 얻은 뒤에 write 메서드를 사용함

- 파일객체 = open(파일이름, 파일모드)
- 파일객체.write('문자열')
- 파일객체.close()

- 다음 내용을 IDLE의 소스 코드 편집 창에 입력한 뒤 실행해보자

file_write_string.py

```
file = open('hello.txt', 'w')    # hello.txt 파일을 쓰기 모드(w)로 열기. 파일 객체 반환
file.write('Hello, world!')      # 파일에 문자열 저장
file.close()                     # 파일 객체 닫기
```

27.1 파일에 문자열 쓰기, 읽기

» 파일에 문자열 쓰기

- 메모장이나 기타 텍스트 편집기를 사용하여 hello.txt 파일을 열어보면 다음과 같이 'Hello, world!' 문자열이 저장된 것을 볼 수 있음

hello.txt

Hello, world!

```
file = open('hello.txt' , 'w' )    # hello.txt 파일을 쓰기 모드(w)로 열기. 파일 객체 반환
```

```
file.write('Hello, world!')        # 파일에 문자열 저장
```

```
file.close()                        # 파일 객체 닫기
```

27.1 파일에 문자열 쓰기, 읽기

▼ 그림 파일에 문자열 쓰기

`file = open()` 파일 열기



`file.write()` 파일 쓰기



`file.close()` 파일 닫기

27.1 파일에 문자열 쓰기, 읽기

» 파일에서 문자열 읽기

- 파일을 읽을 때도 open 함수로 파일을 열어서 파일 객체를 얻은 뒤 read 메서드로 파일의 내용을 읽음
- 단, 이때는 파일 모드를 읽기 모드 'r'로 지정함
 - 변수 = 파일객체.read()
- 다음 내용을 IDLE의 소스 코드 편집 창에 입력한 뒤 실행해보자

file_read_string.py

```
file = open('hello.txt', 'r')    # hello.txt 파일을 읽기 모드(r)로 열기. 파일 객체 반환
s = file.read()                  # 파일에서 문자열 읽기
print(s)                         # Hello, world!
file.close()                     # 파일 객체 닫기
```

실행 결과

Hello, world!

27.1 파일에 문자열 쓰기, 읽기

» 파일에서 문자열 읽기

- 여기서 'r'은 읽다(read)의 r임

```
file = open('hello.txt', 'r')    # hello.txt 파일을 읽기 모드(r)로 열기. 파일 객체 반환
```

```
s = file.read()                  # 파일에서 문자열 읽기  
print(s)                         # Hello, world!
```

```
file.close()                     # 파일 객체 닫기
```

27.1 파일에 문자열 쓰기, 읽기

▼ 그림 파일에서 문자열 읽기

`file = open()` 파일 열기



`file.read()` 파일 읽기



`file.close()` 파일 닫기

27.1 파일에 문자열 쓰기, 읽기

» 자동으로 파일 객체 닫기

- 파이썬에서는 with as를 사용하면 파일을 사용한 뒤 자동으로 파일 객체를 닫아줌

```
with open(파일이름, 파일모드) as 파일객체:  
    코드
```

file_with_as_read_string.py

```
with open('hello.txt', 'r') as file:    # hello.txt 파일을 읽기 모드(r)로 열기  
    s = file.read()                    # 파일에서 문자열 읽기  
    print(s)                           # Hello, world!
```

실행 결과

Hello, world!

- read로 파일을 읽고나서 close를 사용하지 않음
- with as를 사용하면 파일 객체를 자동으로 닫아줌

27.2 문자열 여러 줄을 파일에 쓰기, 읽기

» 반복문으로 문자열 여러 줄을 파일에 쓰기

- 문자열 여러 줄을 쓰려면 간단하게 반복문을 사용하면 됨

file_for_write_string.py

```
with open('hello.txt', 'w') as file:    # hello.txt 파일을 쓰기 모드(w)로 열기
    for i in range(3):
        file.write('Hello, world! {0}\n'.format(i))
```

- py 파일이 있는 폴더의 hello.txt 파일을 열어보면 다음과 같은 내용이 저장되어 있음

```
Hello, world! 0
Hello, world! 1
Hello, world! 2
```

- 파일에 문자열 여러 줄을 저장할 때 주의할 부분은 개행 문자 부분임
- 'Hello, world! {0}\n'와 같이 문자열 끝에 개행 문자 \n를 지정해주어야 줄바꿈이 됨
- 만약 \n을 붙이지 않으면 문자열이 모두 한 줄로 붙어서 저장되므로 주의해야 함

27.2 문자열 여러 줄을 파일에 쓰기, 읽기

» 리스트에 들어있는 문자열을 파일에 쓰기

• 파일객체.`writelines`(문자열리스트)

file_writelines.py

```
lines = ['안녕하세요.\n', '파이썬\n', '코딩 도장입니다.\n']

with open('hello.txt', 'w') as file:    # hello.txt 파일을 쓰기 모드(w)로 열기
    file.writelines(lines)
```

hello.txt

```
안녕하세요.
파이썬
코딩 도장입니다.
```

- `writelines`는 리스트에 들어있는 문자열을 파일에 씀
- `writelines`를 사용할 때는 반드시 리스트의 각 문자열 끝에 개행 문자 `\n`을 붙여주어야 함
- 그렇지 않으면 문자열이 모두 한 줄로 붙어서 저장되므로 주의해야 함

27.2 문자열 여러 줄을 파일에 쓰기, 읽기

» 파일의 내용을 한 줄씩 리스트로 가져오기

- read는 파일의 내용을 읽어서 문자열로 가져오지만 readlines는 파일의 내용을 한 줄씩 리스트 형태로 가져옴

• 변수 = 파일객체.readlines()

file_readlines.py

```
with open('hello.txt', 'r') as file:    # hello.txt 파일을 읽기 모드(r)로 열기
    lines = file.readlines()
    print(lines)
```

실행 결과

```
['안녕하세요.\n', '파이썬\n', '코딩 도장입니다.\n']
```

27.2 문자열 여러 줄을 파일에 쓰기, 읽기

» 파일의 내용을 한 줄씩 읽기

- 만약 파일의 내용을 한 줄씩 순차적으로 읽으려면 `readline()`을 사용함

- 변수 = 파일객체.`readline()`

`file_readlines_while.py`

```
with open('hello.txt', 'r') as file:    # hello.txt 파일을 읽기 모드(r)로 열기
    line = None    # 변수 line을 None으로 초기화
    while line != '':
        line = file.readline()
        print(line.strip('\n'))    # 파일에서 읽어온 문자열에서 \n 삭제하여 출력
```

실행 결과

안녕하세요.
파이썬
코딩 도장입니다.

27.2 문자열 여러 줄을 파일에 쓰기, 읽기

» 파일의 내용을 한 줄씩 읽기

- 파일에 문자열이 몇 줄이나 있는지 모르기 때문에 readline으로 파일을 읽을 때는 while 반복문을 활용해야 함
- while은 특정 조건이 만족할 때 계속 반복하므로 파일의 크기에 상관없이 문자열을 읽어올 수 있음
- readline은 더 이상 읽을 줄이 없을 때는 빈 문자열을 반환하는데, while에는 이런 특성을 이용하여 조건식을 만들어줌
- 즉, `line != ""`와 같이 빈 문자열이 아닐 때 계속 반복하도록 만듦
- 반복문 안에서는 `line = file.readline()`과 같이 문자열 한 줄을 읽어서 변수 line에 저장해주면 됨

27.2 문자열 여러 줄을 파일에 쓰기, 읽기

» 파일의 내용을 한 줄씩 읽기

- 특히 변수 `line`은 `while`로 반복하기 전에 `None`으로 초기화해줌
- 만약 변수 `line`을 만들지 않고 `while`을 실행하면 없는 변수와 빈 문자열 `"`을 비교하게 되므로 에러가 발생함
- `line`을 `None`이 아닌 `"`로 초기화하면 처음부터 `line != "`는 거짓이 되므로 반복을 하지 않고 코드가 그냥 끝나버림

```
line = None    # 변수 line을 None으로 초기화
while line != '':
```

- 문자열을 출력할 때는 `print(line.strip('\n'))`와 같이 `strip` 메서드로 `\n`을 삭제함
- 왜냐하면 파일에서 읽어온 문자열에는 '안녕하세요.\n'와 같이 `\n`이 이미 들어있기 때문임
- 만약 `strip('\n')`을 생략하면 문자열 한 줄을 출력할 때마다 빈 줄이 계속 출력됨
- 문자열 안에 든 `\n`과 `print`가 출력하는 `\n` 때문에 줄바꿈이 두 번 일어남

27.2 문자열 여러 줄을 파일에 쓰기, 읽기

» for 반복문으로 파일의 내용을 줄 단위로 읽기

- 다음은 for 반복문에 파일 객체를 지정하여 줄 단위로 파일의 내용을 읽음

file_for.py

```
with open('hello.txt', 'r') as file:    # hello.txt 파일을 읽기 모드(r)로 열기
    for line in file:                  # for에 파일 객체를 지정하면 파일의 내용을 한 줄씩 읽어서 변수에 저장함
        print(line.strip('\n'))        # 파일에서 읽어온 문자열에서 \n 삭제하여 출력
```

실행 결과

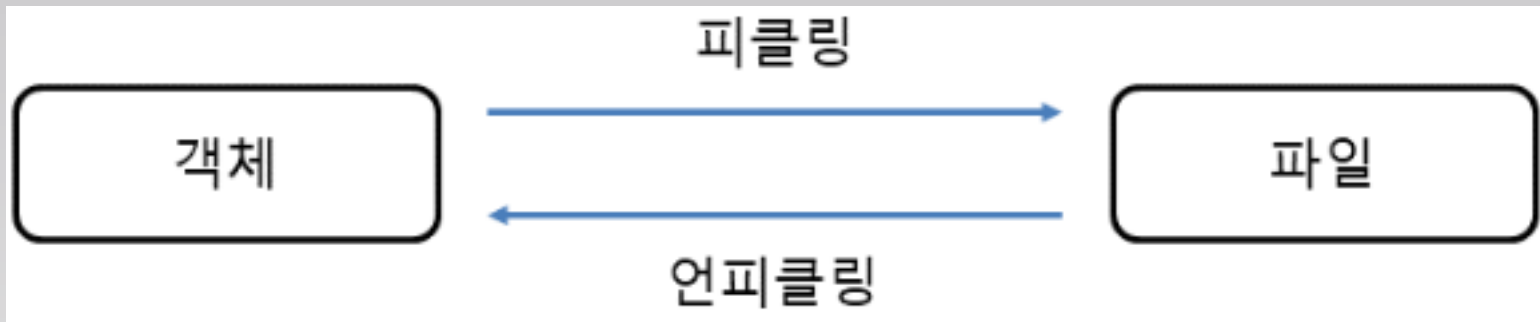
```
안녕하세요.
파이썬
코딩 도장입니다.
```

27.3 파이썬 객체를 파일에 저장하기, 가져오기

» 파이썬 객체를 파일에 저장하기, 가져오기

- 파이썬은 객체를 파일에 저장하는 pickle 모듈을 제공함
- 다음과 같이 파이썬 객체를 파일에 저장하는 과정을 피클링(pickling)이라고 하고, 파일에서 객체를 읽어오는 과정을 언피클링(unpickling)이라고 함

▼ 그림 피클링과 언피클링



27.3 파이썬 객체를 파일에 저장하기, 가져오기

» 파이썬 객체를 파일에 저장하기

- 피클링은 pickle 모듈의 dump 메서드를 사용함

pickle_dump.py

```
import pickle

name = 'james'
age = 17
address = '서울시 서초구 반포동'
scores = {'korean': 90, 'english': 95, 'mathematics': 85, 'science': 82}

with open('james.p', 'wb') as file:    # hello.txt 파일을 바이너리 쓰기 모드(wb)로 열기
    pickle.dump(name, file)
    pickle.dump(age, file)
    pickle.dump(address, file)
    pickle.dump(scores, file)
```

27.3 파이썬 객체를 파일에 저장하기, 가져오기

» 파이썬 객체를 파일에 저장하기

- 소스 코드를 실행하면 .py 파일이 있는 폴더에 james.p 파일이 생성됨
- 확장자를 pickle의 p를 사용했지만 다른 확장자를 사용해도 상관없음
- pickle.dump로 객체(값)를 저장할 때는 open('james.p', 'wb ')과 같이 파일 모드를 'wb'로 지정해야 함
- b는 바이너리(binary)를 뜻하는데, 바이너리 파일은 컴퓨터가 처리하는 파일 형식임
- 메모장 같은 텍스트 편집기로 열어도 사람이 알아보기 어려움
- 지금까지 사용한 .txt 파일은 사람이 알아보기 쉽도록 만든 파일 형식이며 텍스트(text) 파일이라고 부름

27.3 파이썬 객체를 파일에 저장하기, 가져오기

» 파일에서 파이썬 객체 읽기

- 언피클링은 pickle 모듈의 load를 사용함
- 언피클링을 할 때는 반드시 파일 모드를 바이너리 읽기 모드 'rb'로 지정해야 함

pickle_load.py

```
import pickle

with open('james.p', 'rb') as file:    # hello.txt 파일을 바이너리 읽기 모드(rb)로 열기
    name = pickle.load(file)
    age = pickle.load(file)
    address = pickle.load(file)
    scores = pickle.load(file)
    print(name)
    print(age)
    print(address)
    print(scores)
```

실행 결과

```
james
17
서울시 서초구 반포동
{'korean': 90, 'english': 95, 'mathematics': 85, 'science': 82}
```

27.3 파이썬 객체를 파일에 저장하기, 가져오기

» 파일에서 파이썬 객체 읽기

- 마찬가지로 파일에서 객체(값)를 가져올 때도 `pickle.load`를 네 번 사용해야 함
- `name`, `age`, `address`, `scores` 순으로 저장했으므로 가져올 때도 같은 순서로 가져오면 됨

27.3 파이썬 객체를 파일에 저장하기, 가져오기

▼ 그림 파일 모드 조합

