

<제11장. 컴퓨터의 비밀 일기장, 파일 다루기!>

선생님: (손에 두루마리 종이를 들고 나타나며) 애들아, 안녕! 이건 아주 오래된 비밀 두루마리란다. 여기에는 중요한 이야기들이 적혀있지. 컴퓨터 세상에도 이런 두루마리, 즉 **파일**이 있어서 정보를 기록하고 읽어올 수 있단다. 오늘은 이 파일이라는 비밀 일기장을 어떻게 쓰고 읽는지 배워볼 거야!

코딩새싹 🌱: 파일이요? 우리가 그림 파일이나 음악 파일 할 때 그 파일 말씀이세요? 그걸 파이썬으로 만들 수 있다고요? 😬

선생님: 맞아! 그림이나 음악 파일은 조금 복잡하지만, 글자들이 적힌 텍스트 파일은 우리가 직접 만들고 읽을 수 있단다. 마치 우리가 일기를 쓰거나 편지를 쓰는 것처럼 말이지!

<1. 나의 첫 번째 비밀 일기 쓰기: 파일에 글자 저장하기!>

선생님: 먼저, 우리만의 비밀 일기장을 하나 만들어볼까? "테스트.txt"라는 이름의 파일을 만들고, 거기에 글을 써보자.

```
# "테스트.txt" 파일을 쓰기("w") 모드로 열어요. 한글을 위해 encoding="utf8"은 필수!  
file = open("테스트.txt", "w", encoding="utf8")  
file.write("파이썬 코딩 재미있어요.") # 파일에 내용을 적어요.  
file.close() # 다 썼으면 꼭 닫아줘야 내용이 안전하게 저장된답니다!
```

궁금이 💡: open() 함수를 쓰네요! "테스트.txt"는 파일 이름이고, "w"는 뭐예요? 그리고 encoding="utf8"은 왜 있는 거죠?

선생님: 아주 좋은 질문이야, 궁금아!

- "w"는 '쓰기 모드(write mode)'라는 뜻이야. 새 일기장을 펴거나, 기존 일기장이 있다면 내용을 다 지우고 새로 쓰는 것과 같아.
- encoding="utf8"은 우리가 한글처럼 다양한 글자를 파일에 예쁘게 잘 적고 읽기 위한 약속 같은 거란다. 이걸 안 쓰면 한글이 이상하게 보일 수도 있어.
- 그리고 file.close()는 정말 중요해! 일기를 다 쓰고 덮개를 꼭 닫아야 내용이 잘 보관되는 것처럼, 파일도 다 사용하면 꼭 닫아줘야 한단다.

코딩새싹 🌱: 아하! 그럼 이제 "테스트.txt" 파일에 "파이썬 코딩 재미있어요."라고 적힌 거예요? 신기하다!

<2. 일기장 펼쳐보기: 파일 내용 읽어오기!>

선생님: 좋아, 그럼 이제 우리가 내용을 적어둘 "sample.txt"라는 파일을 하나 만들고, 그 내용을 읽어보는 연습을 해볼까? 먼저 "sample.txt"에 내용을 써보자. 이때 "w" 모드는 내용을 싹 지우고 새로 쓴다는 걸 기억해줘!

```
# "sample.txt" 파일을 쓰기 모드로 열고 내용을 작성합니다.  
  
with open("sample.txt", "w", encoding="utf8") as f:  
    f.write("우리나라 좋은 나라") # 첫 번째 줄 내용 (줄바꿈 없음)  
    f.write("빼앗긴 들에도 봄은 오는가?") # 이어서 내용 작성 (줄바꿈 없음)  
  
# with 블록이 끝나면 파일은 자동으로 닫힙니다!
```

궁금이 💡: 어? 선생님, 이번에는 file.close()가 없는데 with open(...) as f:는 뭐예요? 그리고 f.write()를 두 번 했는데, 줄이 바뀌나요?

선생님: 예리한데, 궁금아!

- with open(...) as f: 구문은 파이썬에서 파일을 다루는 아주 현명한 방법이야. 이렇게 쓰면 with 블록(들여쓰기 된 부분)이 끝날 때 파이썬이 **자동으로 파일을 닫아준다**! 그래서 우리가 close()를 깜빡하는 실수를 막을 수 있지. 앞으로 파일을 다룰 땐 이 with 구문을 애용하자!
- 그리고 f.write()를 두 번 연달아 쓰면, 글자들이 그냥 이어서 쪽 적히게 돼. 줄을 바꾸고 싶으면 우리가 직접 줄바꿈 문자 \n을 넣어줘야 한단다. 지금은 "우리나라 좋은 나라빼앗긴 들에도 봄은 오는가?" 이렇게 한 줄로 쪽 이어져 있을 거야.

선생님: 자, 이제 이 "sample.txt" 파일의 내용을 읽어볼까? 이번엔 '읽기 모드("r")'로 열어야 해.

```
# "sample.txt" 파일을 읽기("r") 모드로 열어서 내용을 읽어봅시다.  
  
with open("sample.txt", "r", encoding="utf8") as file:  
    contents = file.read() # 파일의 전체 내용을 하나의 문자열로 읽어와요.  
    print("--- sample.txt 전체 내용 ---")  
    print(contents)  
    print("읽어온 내용의 타입은?", type(contents)) # contents 변수의 자료형 확인!
```

코딩새싹 🌱: file.read() 하니까 파일 안의 모든 내용이 contents 변수 안으로 쏙 들어왔네요! 그리고 type(contents)를 보니까 <class 'str'>! 문자열이군요!

<3. 일기장에 내용 추가하기: 파일 이어쓰기!>

선생님: 맞아. read()는 전체 내용을 하나의 큰 문자열로 가져오지. 그럼 만약 기존 내용은 그대로 두고, 뒤에 새로운 내용을 추가하고 싶으면 어떻게 할까? 그때는 '이어쓰기 모드("a")'를 사용한다. append의 'a'야!

```
# "sample.txt" 파일에 내용을 추가해볼까요? 이어쓰기("a") 모드!

with open("sample.txt", "a", encoding="utf8") as file:

    file.write("\n이거 추가한 거임") # \n으로 줄을 먼저 바꾸고 내용을 추가해요.
```

궁금이 💡: 아! "a" 모드로 열면 기존 내용 뒤에 붙는 거군요! \n은 줄바꿈 문자라고 하셨죠? 그럼 이제 "sample.txt" 파일은 두 줄이 되겠네요?

선생님: 그렇지! 첫 번째 줄은 "우리나라 좋은 나라빼앗긴 들에도 봄은 오는가?" 이고, 두 번째 줄에 "이거 추가한 거임"이 적히게 될 거야.

<4. 한 줄 한 줄 음미하며 읽기: 파일 한 줄씩 읽기!>

선생님: 파일 내용이 아주 길 때는 한 줄씩 차례대로 읽어보는 게 편할 때도 있어. for 반복문을 사용하면 된단다.

```
# "sample.txt" 파일 내용을 한 줄씩 읽어봅시다.

print("\n--- sample.txt 한 줄씩 읽기 ---")

with open("sample.txt", "r", encoding="utf8") as file:

    for line in file: # 파일을 for문에 넣으면 한 줄씩 line 변수에 담아줘요.

        print(line) # 그런데 이렇게 출력하면 줄 사이에 빈 줄이 하나 더 생길 수 있어요!
```

코딩새싹 🌱: 와! for line in file: 이렇게만 해도 한 줄씩 나오는군요! 그런데 정말 줄 사이에 빈 줄이 있는 것 같아요! 왜 그렇죠?

선생님: 좋은 관찰이야! 파일에서 한 줄을 읽어올 때, 각 줄의 끝에는 이미 눈에 보이지 않는 줄바꿈 문자(\n)가 포함되어 있거든. 그런데 print() 함수도 기본적으로 내용을 출력하고 줄을 바꿔주지. 그래서 줄바꿈이 두 번 일어나서 빈 줄이 보이는 거란다. (나중에 line.strip() 같은 걸로 이 눈에 안 보이는 공백이나 줄바꿈 문자를 없앨 수도 있어!)

<5. 나만의 방명록 만들기: 사용자 입력 파일에 저장하기!>

선생님: 자, 이제 우리가 배운 걸로 멋진 프로그램을 만들어볼까? 사용자에게 이름과 이메일을 입력받아서 "user.txt" 파일에 저장하는 방명록 프로그램이야!

```
name = input("이름을 입력하세요: ")
```

```
email = input("이메일을 입력하세요: ")
```

```
with open("user.txt", "w", encoding="utf8") as f: # "w" 모드로 열어서 새로 써요!
```

```
    f.write(f"이름: {name}, 이메일: {email}") # f-string으로 깔끔하게!
```

```
print("\n--- user.txt 내용 확인 ---")
```

```
with open("user.txt", "r", encoding="utf8") as f:
```

```
    print(f.read())
```

궁금이 💡: 와! input()으로 받은 name이랑 email 변수를 f-string을 사용해서 파일에 바로 쓸 수 있네요!

이건 정말 유용하겠어요!

<6. 모든 줄을 리스트로! 그리고 특정 줄만 쏙! readlines()>

선생님: 파일을 읽는 또 다른 방법으로 readlines()가 있어. 이건 파일의 모든 줄을 각각 하나의 문자열로 만들어서, 그 문자열들을 리스트에 담아준다.

```
# "sample.txt" 파일의 모든 줄을 리스트로 읽어오고, 특정 줄을 출력해봅시다.
```

```
# 현재 "sample.txt" 내용은 이렇다고 상상해볼까요?
```

```
# 첫째 줄: 우리나라 좋은 나라빠앗긴 들에도 봄은 오는가?
```

```
# 둘째 줄: 이거 추가한 거임
```

```
print("\n--- sample.txt 모든 줄을 리스트로 읽고, 특정 줄 출력하기 ---")
```

```
with open("sample.txt", "r", encoding="utf8") as file:
```

```
    contents_list = file.readlines() # 모든 줄을 읽어 리스트에 저장!
```

```
    print("readlines()로 읽은 내용의 타입은?", type(contents_list))
```

```
    print("리스트 전체 내용:", contents_list)
```

```
    if len(contents_list) >= 2: # 두 번째 줄이 있는지 확인 (리스트는 0부터 시작!)
```

```
        print("두 번째 줄(인덱스 1) 내용:", contents_list[1])
```

```
    else:
```

```
        print("파일에 두 줄 이상 존재하지 않아요.")
```

코딩새싹 🌱: readlines()를 쓰니까 정말 리스트가 나왔어요! ['우리나라 좋은 나라빠앗긴 들에도

봄은 오는가?\n', '이거 추가한 거임'] 이런 식으로요! 각 줄 끝에 \n도 같이 들어있네요! 그래서 contents_list[1] 하면 두 번째 줄 내용을 꺼낼 수 있는 거군요!

선생님: (흐뭇하게) 아주 잘 이해했구나! 파일을 쓰고, 읽고, 이어 쓰고, 한 줄씩 또는 여러 줄씩 다루는 방법을 배웠으니, 이제 여러분의 프로그램은 훨씬 더 많은 일을 할 수 있게 될 거야! 정보를 저장했다가 다음에 다시 불러오거나, 많은 양의 데이터를 처리하는 것도 가능해지지. 오늘 정말 중요한 마법을 배웠으니, 많이 연습해보렴!