

파이썬 보충 Take 3

파일 입출력

#01. 파일 입출력 객체를 위한 Open 함수

파일 쓰기

```
# 새로운 파일 생성, 혹은 기존의 파일 열기
# -> 파일 경로와 파일 열기 모드 설정
# -> r : 읽기모드
#     w : 쓰기모드 (기존의 내용을 덮어쓴다.)
#     a : 추가모드 (기존의 내용을 유지하고 뒤에 추가)
f = open("helloworld.html", "w", encoding='utf-8')

# 파일에 내용 기록하기
# -> 두 문장을 연결아서 한 줄에 기록한다.
# -> 줄바꿈을 원하는 위치에 줄바꿈을 의미하는 이스케이프 문자(\n)을 사용
f.write("Hello Python!!!\n")
f.write("안녕하세요. 파이썬!!!\n")

# open()함수를 단독으로 사용할 경우 파일작업 완료 후 닫기를 반드시 수행.
f.close()
```

파일 읽기

```
# 이전 예제에서 생성한 파일을 읽어들이기 위해 연다.
# -> 읽기모드(r) 지정에 유의
# -> 인코딩 방식은 저장 방식과 동일하게 지정.(다를 경우 에러)
f = open("helloworld.txt", "r", encoding='utf-8')

# 파일의 모든 내용을 한 번에 문자열로 읽어들인다.
# -> 아래와 같이 파일의 모든 내용을 변수에 저장한 것과 동일한 효과
#     data = "Hello Python!!!\n안녕하세요. 파이썬!!!"
data = f.read()
print(data)

# open()함수를 단독으로 사용할 경우 파일작업 완료 후 닫기를 반드시 수행.
f.close()
```

```
Hello Python!!!
안녕하세요. 파이썬!!!
```

#02. with 블록을 활용한 파일 입출력

파일 쓰기

```
# 이전 예제와 동일하지만 f.close()처리는 자동으로 수행함.
with open("hellopython.txt", "w", encoding='utf-8') as f:
    # with 블록 안에서만 파일 객체 f가 유효함
    for i in range(0, 10):
        f.write("%d >> " % i)
        f.write("Life is too short, ")
        f.write("you need python\n")

    print("파일 저장이 완료되었습니다.")
```

파일 저장이 완료되었습니다.

파일 읽기

```
# 읽기 모드로 파일 객체 생성하기
# `w`에서 `r`로 모드가 변경되어야 하므로 파일 객체를 새로 생성해야 한다.
with open("hellopython.txt", "r", encoding='utf-8') as f:
    # 파일의 내용을 변수에 저장함
    data = f.read()
    print(data)
```

```
0 >> Life is too short, you need python
1 >> Life is too short, you need python
2 >> Life is too short, you need python
3 >> Life is too short, you need python
4 >> Life is too short, you need python
5 >> Life is too short, you need python
6 >> Life is too short, you need python
7 >> Life is too short, you need python
8 >> Life is too short, you need python
9 >> Life is too short, you need python
```

파일의 내용을 한 줄씩 읽기

```
with open("helloworld.txt", "r", encoding='utf-8') as f:
    # 파일의 각 행을 원소로 갖는 리스트 생성
    # -> 아래의 구문과 동일한 효과
    #     lines = ["Hello Python!!!\n", "안녕하세요. 파이썬!!!\n"]
    lines = f.readlines()
    print(lines)
    print("-" * 30)

    # 전체 리스트의 크기 확인
    size = len(lines)
    print("읽어들이는 데이터는 총 %d줄 입니다." % size)
    print("-" * 30)

    for item in lines:
```

```
# 읽어들이는 데이터에 줄바꿈 문자가 포함되어므로 공백제거 필요
print( item.strip() )
```

```
['Hello Python!!!\n', '안녕하세요. 파이썬!!!\n']
```

읽어들인 데이터는 총 2줄 입니다.

Hello Python!!!

안녕하세요. 파이썬!!!