
파일

파일 입출력

❖ 파일 예외 처리

```
try:
    f = open("live.txt", "rt")
    text = f.read()
    print(text)
except FileNotFoundError:
    print("파일이 없습니다.")
finally:
    f.close()    # 파일 열기에 실패한 경우 f가 정의되지 않아 또 예외 발생
```

파일 입출력

❖ with ~ as 문 (컨텍스트 매니저)

- open() 함수와 함께 with ~ as문을 사용하면 명시적으로 close() 함수를 호출하지 않아도 파일이 항상 닫힘.

with open(파일이름) as 파일객체:

코드블록
이곳에서 읽거나
쓰기를 한 후
그냥 코드를 빠져나감.

"파일객체 = open(파일이름)"
과 동일

With에 의해 코드 블록을 벗어
날 때 자동 close()

❖ 예제

```
with open('test.txt', 'r') as file:  
    str = file.read()  
    print(str)  
#file.close()
```

파일 입출력

❖ 파일 예외 처리

```
try:
    with open('test.txt', 'r') as file:
        text = file.read()
        print(text)
except Exception as e:
    print(e)
```

파일 입출력

❖ pickle 모듈

- 파이썬 자료형 그대로 저장하고, 그대로 로드(복원)
- 반드시 binary 모드로 오픈해야 함
- 다른 언어와 호환성은 없음
- import pickle
- 저장하기
 - `pickle.dump(data, file)`
 - data: 저장할 데이터
 - file: "bw"로 오픈한 파일 객체
- 로드하기
 - `data = pickle.load(file)`
 - file: "br"로 오픈한 파일 객체
 - data: 복원한 데이터

파일 입출력

❖ pickle 모듈(저장하기)

```
def save(fpath, data):  
    try:  
        with open(fpath, 'wb') as file:  
            pickle.dump(data, file)  
    except Exception as e:  
        print(f"{fpath} 파일 쓰기에 실패했습니다.")  
        print(e)
```

파일 입출력

❖ pickle 모듈(불러오기)

```
def load(fpath):  
    try:  
        with open(fpath, 'rb') as file:  
            data = pickle.load(file)  
            return data  
    except:  
        print(f"{fpath} 파일 읽기에 실패했습니다.")
```