Ch4. 영속성: 파일에 데이터 저장하기

- summarized by 전승일(2014/)
- source code: http://www.headfirstlabs.com/books/hfpython/
- support site: http://python.itcarlow.ie

프로그램은 데이터를 만듭니다

- 142쪽 그림 참조
- 실습(file read): 대사를 man과 other라는 list에 각각 나누어 저장한 후 화면에 출력하기

쓰기 모드로 파일 열기

- 실습(file write mode): 화면에 출력하는 대신 파일에 쓰기

예외가 발생한 후에도 파일은 여전히 열려 있습니다

- 만일 파일을 쓰다가 에러가 발생하면, 예외가 발생하고, 파일이 열려진 채로 방치되어 문제 발생 (150쪽 그림 참조)
- file은 열면 반드시 닫도록 해야 함

try 문을 finally로 확장하기

- try/except/finally
- 실습(finally): finally는 항상 수행 됨: 단순히 close를 finally 안에 넣으면 문제 해결될까?

에러의 종류를 아는 것만으로는 충분하지 않습니다

- 에러의 유형과 더불어 상세한 내용을 알 필요가 있음

```
except IOError as err:
    print('File error: ' + str(err))
```

- 실습(finally2)

with를 사용해서 파일 작업하기

- 파일이 열리지 못하고 실패한 경우는 그대로 예외처리를 하고, 파일이 열린 후에 실패한 경우 파일을 닫고 예외처리 해 주는 기능이 있다면... => 콘텍스트 관리 프로토콜
- 실습(with write): with 구문 sample
- 실습(with write2): finally2 프로그램을 with 구문으로 수정

기본 형식이 파일에 맞지 않네요

- print는 python data를 사람이 보기 편하게 출력할 뿐
- print로 출력된 결과를 다시 python data로 불러들이기는 매우 어려움 (data를 parsing 해야 함)

print_lol()을 수정하는 게 어떨까요?

- data를 parsing 하기 좀 편하게 print 하는 방법 (한계가 있음)
- 실습(print_lol): print 시에 file로 보내는 인자 추가

데이터를 pickle하기

- 데이터를 나중에 python으로 읽어들이기 편하게 하려면 pickle 모듈을 사용!

```
import pickle
```

- 개념: 168쪽 그림 참조

dump로 저장하고 load로 읽습니다

- pickle.dump(object, file_handler)
- object = pickle.load(file_handler)
- pickle.dump는 binary file로 저장 함: open 함수 호출 시, "rb" 혹은 "wb" 사용
- pickle의 예외처리는 pickle.PickleError 유형을 사용 함
- 실습(pickle_test): 앞의 예제를 pickle을 사용하여 수정
- 실습(An IDLE Session): 저장된 파일의 데이터 확인 하기

pickle을 이용한 범용 파일 I/O가 최고입니다!

Additional Information

- another good book: http://www.pythonlearn.com