

파이썬 프로그래밍

파일 입출력



한국기술교육대학교
온라인평생교육원

■ 파일로의 지속 모듈

1~2. 지속성과 지속성 기능을 지원하는 모듈

- 지속성(Persistence)
 - 프로그램 내에 생성된 각종 객체들을 해당 프로그램 종료 이후에도 존재하게 만들고, 그것들을 동일한 또는 다른 프로그램에서 사용하는 기능
- 지속성 기능을 지원하는 모듈
 - DBM 관련 모듈
 - * anydbm, dbm, gdbm, dbhash, dumbdbm
 - * anydbm: 시스템에서 사용가능한 모듈 중 가장 최적의 모듈을 반환함
 - 기본적으로 dumbdbm을 반환한다
 - 사전 자료형을 사용하는 것과 동일한 방법으로 사용
 - pickle 모듈
 - * 파이썬의 객체를 저장하는 일반화된 지속성 모듈
 - * 파이썬의 기본 객체뿐만 아니라 사용자 정의의 복잡한 객체도 저장 가능
 - * 기본적으로 텍스트 모드로 저장하지만 이진 모드로도 저장 가능

- 파이썬에서는 모든 것이 객체인데 메모리에 존재
- 메모리에 존재하기 때문에 프로그램 종료 시 삭제됨
- 지속성 관련된 모듈 활용 시 객체 파일 저장 가능
- 얼음과 같이 객체를 얼려 언제든지 사용 가능 = '지속성'
- DBM 관련 모듈 → 파이썬이 제공하는 내장자료형 저장에 최적화
- pickle 모듈 → 파이썬 내장자료형 + 사용자가 정의한 객체 저장 가능
- DBM 모듈보다는 pickle 모듈이 좀 더 일반적

■ 파일로의 지속 모듈

1~2. 지속성과 지속성 기능을 지원하는 모듈

- 피클링(pickling) 모듈 사용하기

```
import pickle

phone = {'tom':4358382, 'jack':9465215, 'jim':6851325, 'Joseph':6584321}
List = ['string', 1234, 0.2345]
Tuple = (phone, List) # 리스트, 튜플, 사전의 복합 객체

f = open('pickle.txt', 'w') # 파일 객체를 얻는다.

pickle.dump(Tuple, f) # 파일로 출력(pickling), 복합 객체 출력
f.close()

f = open('pickle.txt')

x,y = pickle.load(f) # 파일에서 읽어오기. 튜플의 내용을 x, y에 받는다.
print x # 사전
print y # 리스트
```

```
{'jim': 6851325, 'Joseph': 6584321, 'jack': 9465215, 'tom': 4358382}
['string', 1234, 0.2345]
```

- dump(Tuple, f) → 저장하고자 하는 Tuple을 f에 넣어 열림
- dump() 함수의 반대 → load() 함수

■ 파일로의 지속 모듈

1~2. 지속성과 지속성 기능을 지원하는 모듈

```
import pickle

class Simple: # 가장 단순한 클래스를 정의
    pass

s = Simple() # 인스턴스 객체 생성
s.count = 10 # 인스턴스 이름 공간에 변수 생성

f = open('pickle2.txt', 'w')
pickle.dump(s, f) # 인스턴스 저장
f.close()

f = open('pickle2.txt')
t = pickle.load(f) # 인스턴스 가져오기
print t.count
```

10

- class 정의 방법 → 'class' 키워드 사용
- 콜론(:)을 썼으므로 그 아래 몸체를 적겠다는 것
- pass → 몸체 적지 않고 끝내겠다는 의미
- s = Simple() → Simple 클래스가 가지고 있는 생성자 호출
- s.count = 10 → 인스턴스 내 새로운 변수를 정의
- pickle.dump(저장하고자 하는 객체, 저장하는 위치)
- pickle.load(저장했던 위치) → 저장했던 객체 불러옴