

Ch9. 데이터 관리: 입력 처리하기

- summarized by 전승일(2014/12/01)
- source code: <http://www.headfirstlabs.com/books/hfpython/>
- source code: \$ git clone <https://github.com/sijeon/HeadFirstPython>
- support site: <http://python.itcarlow.ie>

여러분의 선수 기록 앱이 전국적으로 사용되고 있어요

- 선수 데이터에 새로운 기록을 추가하는 기능 필요

입력을 받기 위해서는 FORM이나 다이얼로그를 사용하세요

HTML FORM 템플릿 만들기

- templates/form.html에 템플릿 작성
- cgi-bin/yate.py에 create_inputs, do_form 함수 추가
- cgi-bin/test_form.py로 테스트 하기: http://localhost:8080/cgi-bin/test_form.py

데이터가 여러분의 CGI 스크립트에 도착했습니다

- cgi-bin/add_timing_data_test.py를 통해서 CGI FORM으로 전달된 데이터를 처리하기

안드로이드 폰에서 입력하도록 요청하기

- dialogGetInput 메소드를 이용하여 사용자 입력값을 변수에 저장하기: 340쪽 코드 참조
- urllib2 모듈의 urlopen을 이용하여 POST 하기: 342쪽 코드 참조
- 실행하기: 343쪽 실행화면 참조, web과 동일

서버 데이터를 갱신하도록 만들 차례입니다

- text 파일을 갱신해야 하는가, pickle 파일을 갱신해야 하는가? 344쪽 그림 참조
- pickle을 갱신하면, put_to_store()가 실행될 때 데이터가 사라지게 됨
- text 파일을 갱신하면, pickle로 저장하기 전에 어디선가 get_from_store()를 호출하면 예전 데이터가 사용 됨

경쟁 조건 피하기

- 짧은 시간이긴 하지만 두 데이터가 불일치하게 되는 상황: race condition
- race condition으로 문제가 발생하면 debug가 힘들다... race condition을 피할 것!
- 한 사용자가 데이터를 사용한다면 문제가 없겠지만, 여러 사용자가 동시에 접속하여 사용하고 갱신한다면.... race condition으로 문제 발생

더 나은 데이터 저장 메커니즘이 필요합니다.

- 데이터를 2군데(text file과 pickle 파일)에서 관리하는 것이 근본 문제

데이터베이스 관리 시스템 사용하기

- Oracle DBMS, PostgreSQL, MS SQL, MySQL, MariaDB, ...
- 간단한 DBMS?

Python에는 SQLite가 들어 있습니다

- import sqlite3

- 기본 내장 모듈, 단 OS에 sqlite가 깔려 있어야 한다 (android, OS X 등에는 기본적으로 설치되어 있음)

Python Database API 사용하기

- 연결 -> 생성(커서) -> 조작 -> 커밋 or 롤백 -> 닫기
- 350쪽 그림 참조

데이터베이스 API를 Python 코드로

```
import sqlite3

connection = sqlite3.connect('test.sqlite')

cursor = connection.cursor()

cursor.execute("""SELECT DATE('NOW')""")

connection.commit()

connection.close()
```

간단한 데이터베이스 설계에도 할 일이 많습니다

- 현재 데이터의 상태: 352쪽 그림 참조

데이터베이스 schema 정의하기

- 353쪽 그림 참조

데이터가 어떻게 보이나요?

- 354쪽 그림 참조
- DB 생성하기: createDBtables.py

pickle에서 SQLite로 데이터 이전하기

- initDBtables.py

어느 선수가 어떤 ID를 갖고 있나?

```
cursor.execute("SELECT id from athlete WHERE name=? AND dob=?", (name, dob))
```

- cursor.fetchone() : 1행 반환 as list
- cursor.fetchmany(): 여러 행 반환 as list
- cursor.fetchall() : 모든 행 반환 as list

시간 데이터 추가하기

- initDBtables.py

SQLite 데이터 관리 도구

- 명령행 도구

```
sqlite3  
.help  
.quit
```

- GUI 툴 이용하기

```
SQLite Pro free on AppStore
```

SQLite와 기존 Webapp 통합하기

- MVC에서 Model, 즉 Data Model 관련 코드 수정
- athletemodel.py
- API 변경에 따른 기존 코드 수정: 계속 이용할 수 있는 부분 없음: 새로 작성

그래도 선수 명단은 필요합니다

- get_names_from_store() 작성

ID를 사용해서 선수 정보 가져오기

- get_athlete_from_id() 작성
- 선수 이름과 동시에 ID값 가져오기: get_namesID_from_store() 작성
- 기존 CGI 수정

```
generate_list.py 수정
generate_timing_data.py 수정
```

- 안드로이드용 app을 위한 CGI 추가 수정

```
generate_list.py 수정
generate_timing_data.py 수정
generate_names.py 수정
generate_data.py 수정
```

안드로이드 앱도 수정해야 합니다.

- coachapp.py 수정
- get2inputapp.py 작성

SQLite의 선수 데이터 갱신하기

- cgi-bin/add_timing_data.py 수정 : 안드로이드에서 입력한 데이터가 SQLite에 반영 됨

NUAC은 열광의 도가니가 되었습니다!

- SQL을 통해 자유자재로 데이터를 분석할 수도 있음

Additional Information
