Using DB in Python 추가자료

2013 Spring SPARCS Freshman

SPARCS 12’ 이태현(whitegold)

Sqlite3 간단한 이용법

“DROP TABLE IF EXIST Human”

Human 이라는 Table 존재? 없애버리자

“CREATE TABLE Human(id INT, Name TEXT)”

Human(int type id, TEXT type Name) Table 생성

“INSERT INTO Human VALUES(1,“grandmar..”)”

Human 이란 Table에 (1,”grandmar..”)라는 데이터를 추가하자

Sqlite3 간단한 이용법

“SELECT * FROM Human”

Human 에 있는 모든 데이터 가져오기

“DELETE FROM Human”

Human에 있는 모든 데이터 삭제

“SELECT * FROM Human where id==1”

Human 에 있는 데이터 중 id가 1인 data들을 가져온다

.quit

나가기

Executing sqlite3 in Shell

- Shell 에 sqlite3 **.db

```
whitegold@bit:~/flask_st$ sqlite3 example.db
SQLite version 3.7.9 2011-11-01 00:52:41
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite>
```

- create table human(id int, name text) ;
  - 세미콜론 잊지 말 것

```
whitegold@bit:~/flask_st$ sqlite3 example.db
SQLite version 3.7.9 2011-11-01 00:52:41
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> create table human(id int, name text);
```

Executing sqlite3 in Shell

- INSERT INTO human VALUES(1,”whitegold”);

```
sqlite> INSERT INTO human VALUES(1,"whitegold");
sqlite>
```

- create table human(id int, name text) ;

```
sqlite> select * from human;
1|whitegold
sqlite>
```

- delete from human;

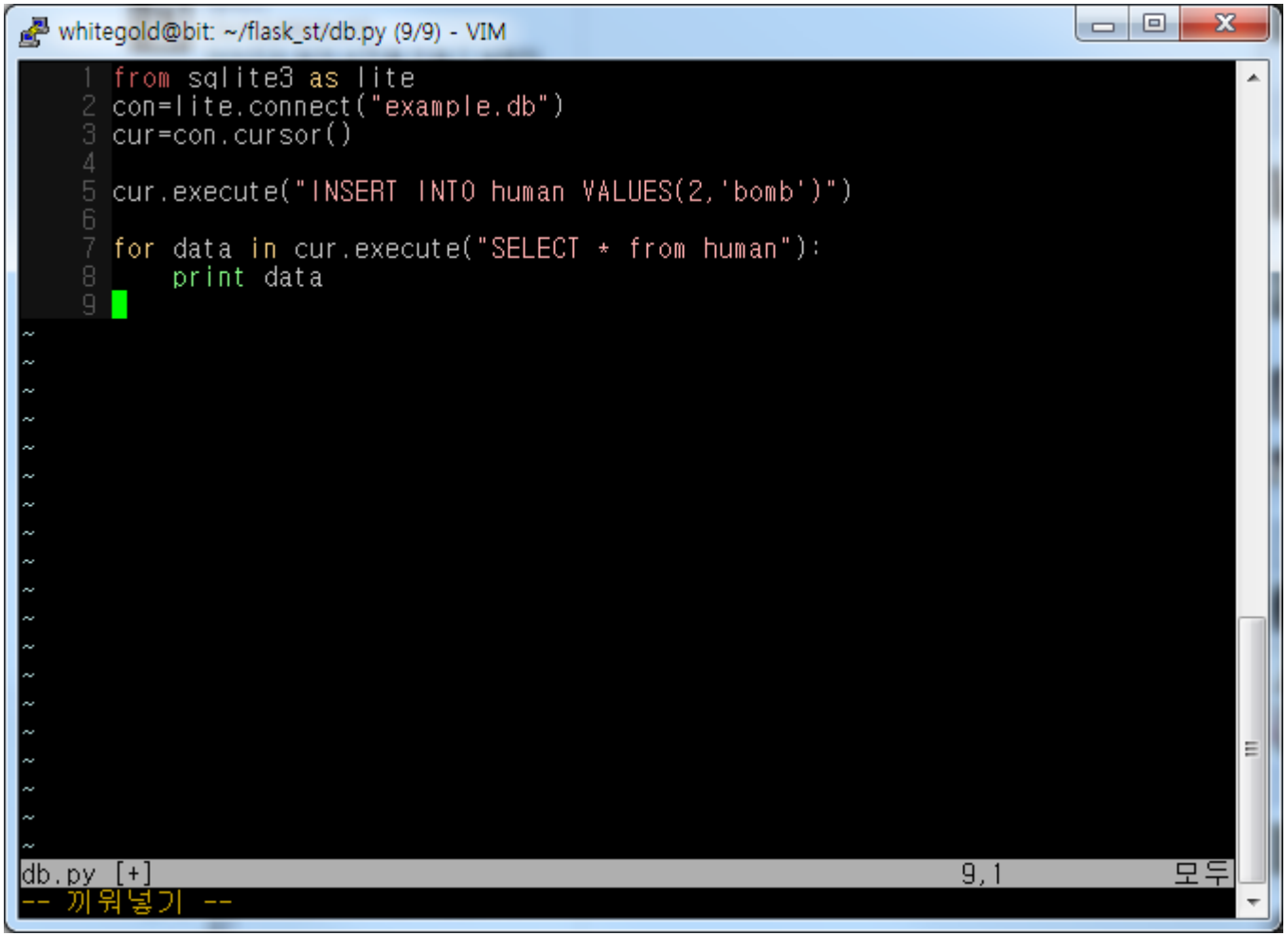
```
sqlite> select * from human;
1|whitegold
sqlite> delete from human;
sqlite> select * from human;
sqlite>
```

- .quit

Using Sqlite3 in FLASK

- 주의
  - CREATE TABLE 은 미리 Shell 에서 해놓기
  - connect, cursor 생성은 함수 안에서
    - thread 끼리의 충돌이 생길 수 있다.
  - 다 이용했으면 close() 를 하자

Executing sqlite3 in Python



Executing sqlite3 in FLASK

whitegold@bit: ~/flask_st/db.py (14/29) - VIM

```python
from flask import Flask
from flask import render_template
from flask import request,url_for,redirect
import sqlite3 as lite

app=Flask(__name__)
num=-1
name=""
@app.route('/db')
def usingdb():
    con=lite.connect('example.db')
    cur=con.cursor()
    cur.execute("INSERT INTO human VALUES(%d,'%s')"%(num,name))
    cur.execute("SELECT * FROM human")
    tries = cur.fetchall()
    con.commit()
    cur.close()
    con.close()
    return render_template('db.html',tries=tries)

@app.route("/submit",methods=['GET','POST'])
def submit():
    global num,name
    num=int(request.form['number'])
    name=str(request.form['name'])
    return redirect(url_for('usingdb'))

app.run(host='0.0.0.0',port=20780,debug=True)

```

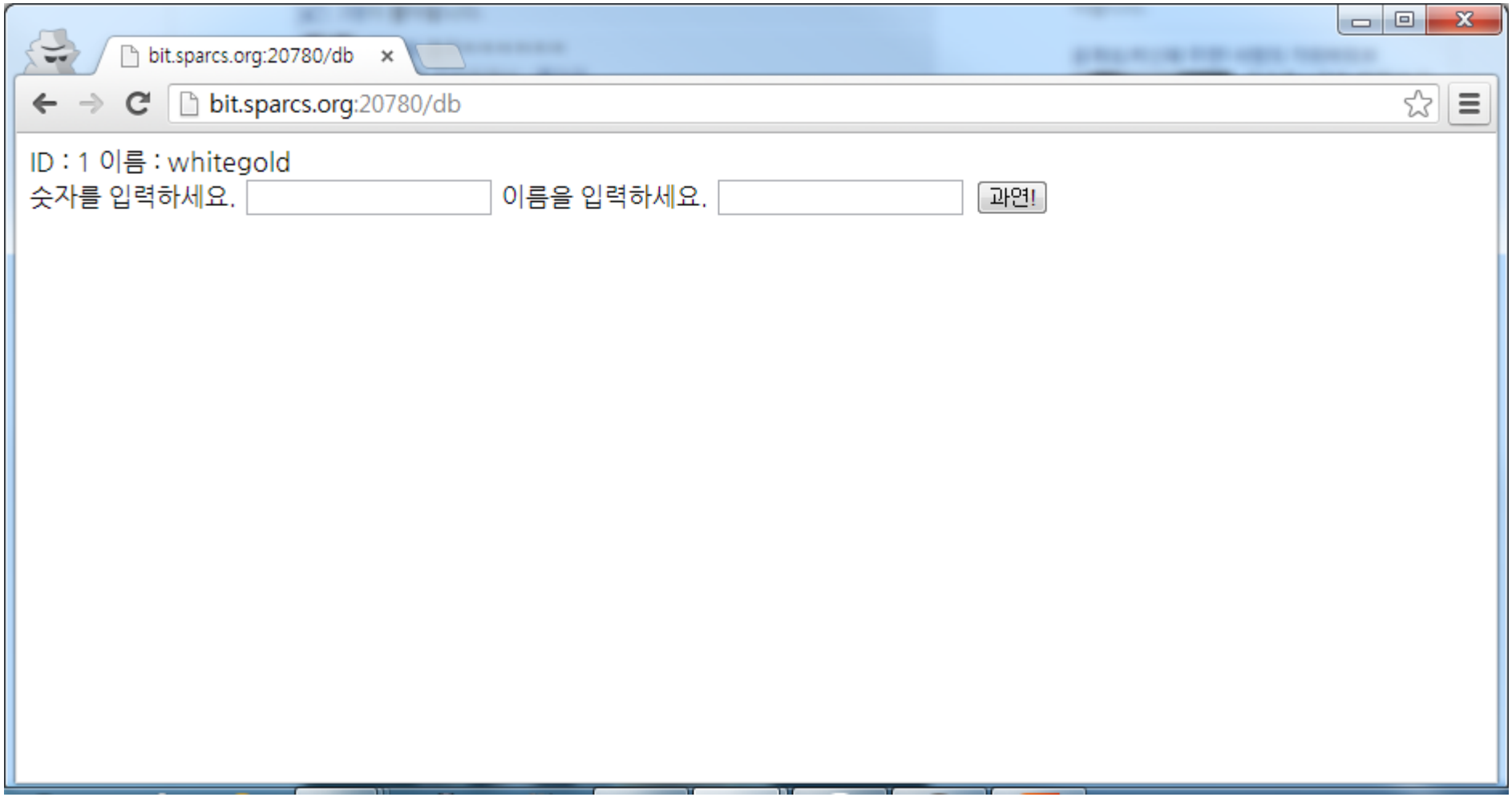
whitegold@bit: ~/flask_st/templates/db.html (6/11) - VIM

```html
<!doctype html>

{% for i in tries %}
ID : {{i[0]}} 이름 : {{i[1]}} <br>
{% endfor %}

<form action="/submit" method="POST">
    숫자를 입력하세요. <input type="text" name="number" />
    이름을 입력하세요. <input type="text" name="name" />
    <input type="submit" value="과연!" />
</form>
```

Executing sqlite3 in FLASK



RESET?

whitegold@bit: ~/flask_st/db.py (40/40) - VIM

```python
from flask import Flask
from flask import render_template
from flask import request,url_for,redirect
import sqlite3 as lite

app=Flask(__name__)
num=-1
name=""
@app.route('/db')
def usingdb():
    con=lite.connect('example.db')
    cur=con.cursor()
    if not num==-1:
        cur.execute("INSERT INTO human VALUES(%d,'%s')"%(num,name))
    cur.execute("SELECT * FROM human")
    tries = cur.fetchall()
    con.commit()
    cur.close()
    con.close()
    return render_template('db.html',tries=tries)

@app.route("/submit",methods=['GET','POST'])
def submit():
    global num,name
    num=int(request.form['number'])
    name=str(request.form['name'])
    return redirect(url_for('usingdb'))

@app.route("/reset", methods=['GET','POST'])
def reset():
    con=lite.connect('example.db')
    cur=con.cursor()
    cur.execute("DELETE from human")
    num=-1
    con.commit()
    cur.close()
    con.close()
    return redirect(url_for('usingdb'))
app.run(host='0.0.0.0',port=20780,debug=True)
```

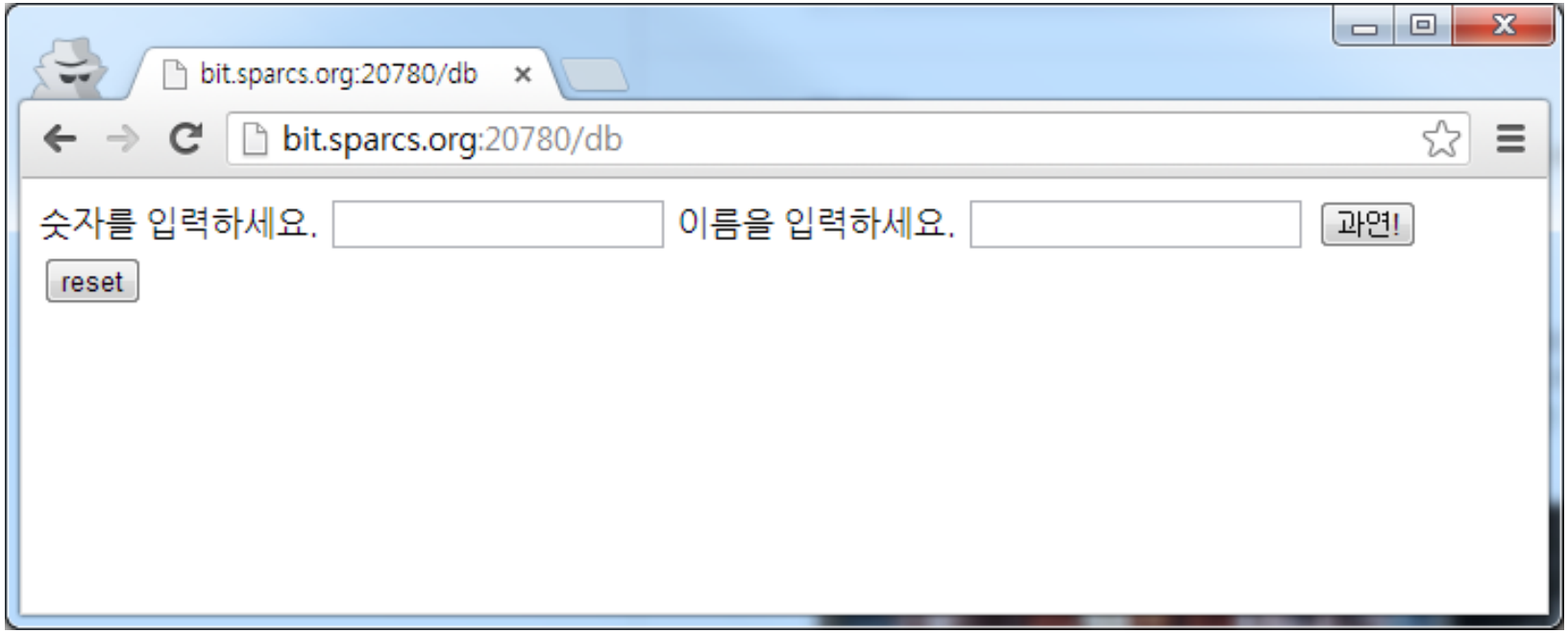
```html
<!doctype html>

{% for i in tries %}
ID : {{i[0]}} 이름 : {{i[1]}} <br>
{% endfor %}

<form action="/submit" method="POST">
    숫자를 입력하세요. <input type="text" name="number" />
    이름을 입력하세요. <input type="text" name="name" />
    <input type="submit" value="과연!" />
</form>

<form action="/reset" method="POST">
    <input type="submit" value="reset" />
</form>
```

RESET?


bit.sparcs.org:20780/db
bit.sparcs.org:20780/db
숫자를 입력하세요.
이름을 입력하세요.
과연!
reset

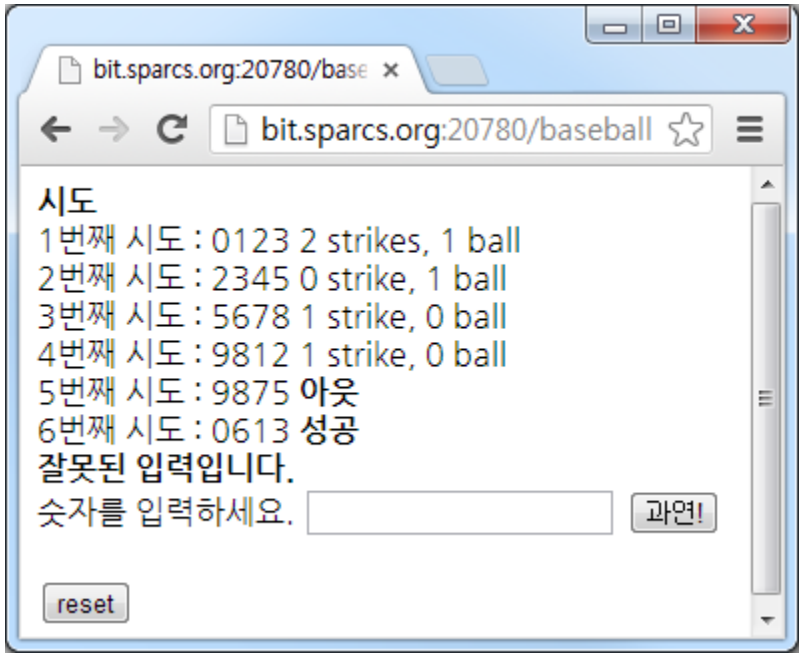
실습 2 : Baseball Game(숫자야구) Upgrade

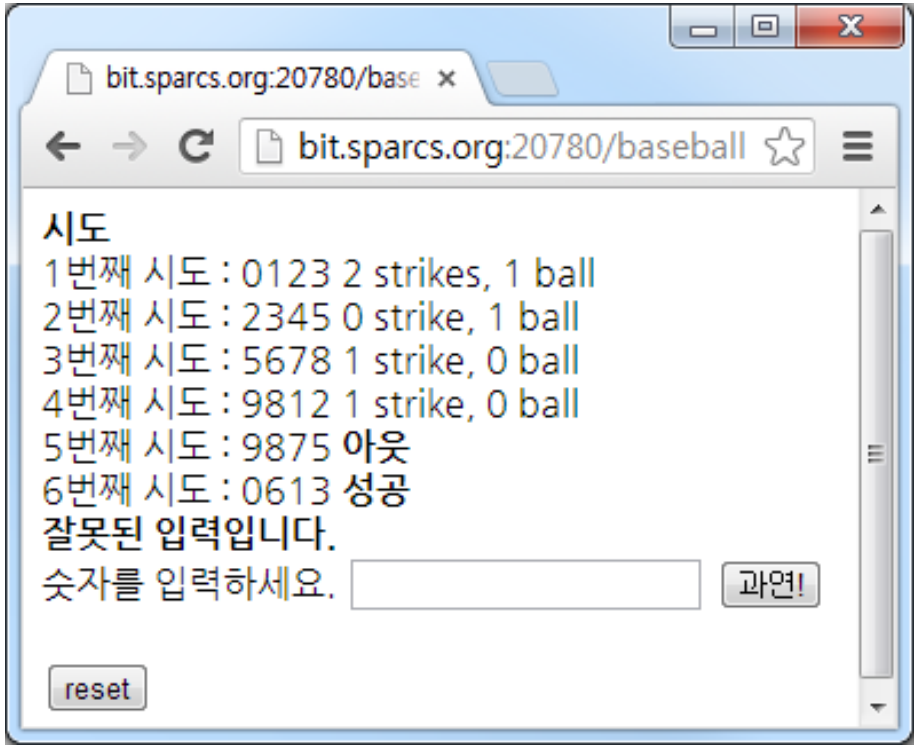
- baseball.py baseball.html
- baseball.db

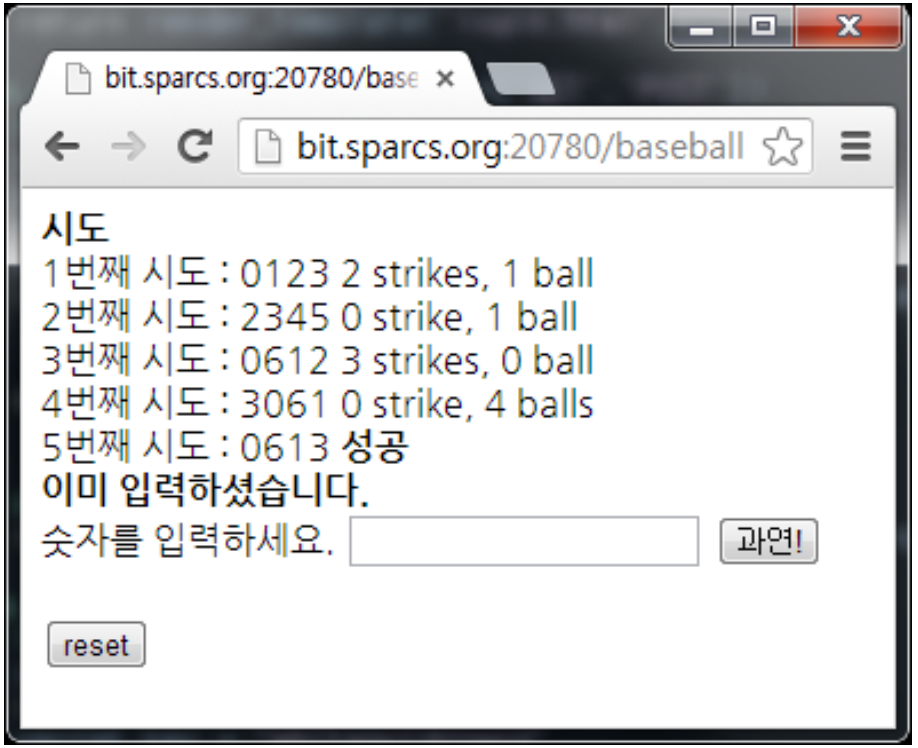
- Conditions

- 여태까지 시도했던 숫자들을 db에 저장
  (단, 실패한 것은 저장하지 않는다)
- Template에서 숫자를 입력할 수 있게 text 를 만든다
- 모든 db가 날아갈 수도 있게 reset 버튼을 만든다
- 답은 py에서 상수로
- 이미 입력했던 숫자, 불가능한 string 이 입력됐을 시 표시

답이 0613 일 때




bit.sparcs.org:20780/base
bit.sparcs.org:20780/baseball
시도
1번째 시도 : 0123 2 strikes, 1 ball
2번째 시도 : 2345 0 strike, 1 ball
3번째 시도 : 5678 1 strike, 0 ball
4번째 시도 : 9812 1 strike, 0 ball
5번째 시도 : 9875 아웃
6번째 시도 : 0613 성공
잘못된 입력입니다.
숫자를 입력하세요.
과연!
reset


bit.sparcs.org:20780/base
bit.sparcs.org:20780/baseball
시도
1번째 시도 : 0123 2 strikes, 1 ball
2번째 시도 : 2345 0 strike, 1 ball
3번째 시도 : 0612 3 strikes, 0 ball
4번째 시도 : 3061 0 strike, 4 balls
5번째 시도 : 0613 성공
이미 입력하셨습니다.
숫자를 입력하세요.
과연!
reset

끝

수고하셨습니다 ☺