

프로그래밍언어 개론  
- HW04 -  
Cute17 Scanner

제출일자	2017.04.03.
분 반	01
이 름	신종욱
학 번	2013024223

### 1.구현방법

pdf파일에 있는 mDFA에 나와있는 현 State와 Input에 따른 State 변화를 보면서 transM함수를 써서  
현 State와 Input에 따른 다음 상태를 숫자로 표현해주어서 상태도를 완성하고  
그 외 키워드로 이루어진 문자열이라면 새로운 상태로 정의해주기위해 check keyword함수로 단어를 체크하고  
키워드일 경우에는 알맞은 상태(숫자)를 리턴하여서 Type을 표시하였다.

### 2. 느낀점

Pdf에 나와있는 함수설명을 읽고 파일을 디버깅해보니 프로그램의 구조를 쉽게 이해 할 수 있어서 쉽게 완성하였다.  
설명이 없는 Scanner Class를 해석하는데 시간이 걸렸었다. 결국엔 문자열을 쪼개면서 스캔하는 것 이다

### 3. 실행결과

```
209 def tokenize(self):
210     print "=== tokenize ==="
211
212     def build_token(type, lexeme): return Token(type, lexeme)
213     print self.source
214     cute_scanner = Scanner(self.source)
215     return cute_scanner.scan(self.transM, build_token)
216
217
218 test_cute = CuteScanner("Test car + ' - * #T ( ) eq?")
219 test_tokens = test_cute.tokenize()
220 print test_tokens
```

Run hw03.Frame

```
C:\Python27\python.exe "C:/Users/신종욱/Downloads/hw03/hw03.Frame.py"
=== tokenize ===
Test car + ' - * #T ( ) eq?
scanning...
[ID: Test]
[CAR: car]
[PLUS: +]
[APOSTROPHE: ']
[MINUS: -]
[TIMES: *]
[TRUE: #T]
[L_PAREN: (]
[R_PAREN: )]
[EQ_Q: eq?]
[[ID: Test], [CAR: car], [PLUS: +], [APOSTROPHE: '], [MINUS: -], [TIMES: *], [TRUE: #T], [L_PAREN: (], [R_PAREN: )], [EQ_Q: eq?]]
```

Program finished with quit code 0

```
LT = 12
GT = 13
EQ = 14
💡 APOSTROPHE = 15
#QUESTION=16

DEFINE = 20
LAMBDA = 21
```

question이 없어서 테스트에 영향은 안 주도록 주석처리로만 만들어놨습니다