연습문제 9장

1

In [1]:
```
ws = []
for line in open('test.txt'):
    if line.startswith('#'):
        continue
    ws.extend(line.split())

print len(ws)
```

```
10
```

2

In [2]:
```
# 방법 1
with open('number.txt', 'w') as f:
    for k in range(10):
        print >> f, k
```

In [3]:
```
# 방법 2
from __future__ import print_function

with open('number.txt', 'w') as f:
    for k in range(10):
        print(k, file=f)
```

3

In [4]:
```
%%file s.txt
pig ham
cat dog
ham bird
dog pig
```

```
Writing s.txt
```

In [5]:
```
for line in sorted(open('s.txt')):
    print(line.rstrip())
```

```
cat dog
dog pig
ham bird
pig ham
```

4

In [6]:
```
for line in sorted(open('s.txt'), key=lambda e: e.split()[1]):
    print(line.rstrip())
```

```
ham bird
cat dog
pig ham
dog pig
```

5

In [7]:
```
ws = open('s.txt').read().split()
for i in range(0, len(ws), 3):
    print(' '.join(ws[i:i+3]))
```

```
pig ham cat
dog ham bird
dog pig
```

6

다음 코드를 mygrep.py 이름으로 저장하고 다음 예와 같이 실행한다.

```
python mygrep.py pig *.txt
```

In [8]:
```
import glob
import sys

args = sys.argv[1:]
search = args[0]
fpatt = args[1]

for fpath in glob.glob(fpatt):
    for n, line in enumerate(open(fpath)):
        if search in line:
            print fpath, n+1, line.strip()
```

7

다음 코드를 mygrep2.py 이름으로 저장하고 다음 예와 같이 실행한다.

```
python mygrep2.py p?g *.txt
```

In [9]:
```
import glob
import sys
import re

args = sys.argv[1:]
search = args[0]
fpatt = args[1]

for fpath in glob.glob(fpatt):
    for n, line in enumerate(open(fpath)):
        if re.search(search, line):
            print fpath, n+1, line.strip()
            print
```

In [10]:
```
re.search(search, line)
```

8

In [11]:
```
import time

time.strftime('%Y%m%d %H%M%S')  # 현재 시간
```

Out[11]: '20160118 074333'

In [12]:
```
def add_log(ip, access_page, access_time):
    out_line = '{}:{}:{}\n'.format(ip, access_page, access_time)
    with open('Weblog', 'a') as f:
        f.write(out_line)
```

In [13]:
```
import time

ip = '128.134.45.23'
access_page = '/gallery/index.html'

for i in range(10):
    access_time = time.strftime('%Y%m%d %H%M%S')
    add_log(ip, access_page, access_time)
    time.sleep(1)
```

In [14]:
```
import time

ip = '128.134.45.24'
access_page = '/gallery/memory.html'

for i in range(5):
    access_time = time.strftime('%Y%m%d %H%M%S')
    add_log(ip, access_page, access_time)
    time.sleep(1)
```

9

In [15]:
```
# 라인단위로 읽어 항목을 구분한다

[tuple(line.split(':')) for line in open('Weblog')]
```

Out[15]:
```
[('128.134.45.23', '/gallery/index.html', '20160118 074558\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074559\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074600\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074601\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074602\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074603\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074604\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074605\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074606\n'),
 ('128.134.45.23', '/gallery/index.html', '20160118 074607\n'),
 ('128.134.45.24', '/gallery/memory.html', '20160118 074608\n'),
 ('128.134.45.24', '/gallery/memory.html', '20160118 074609\n'),
 ('128.134.45.24', '/gallery/memory.html', '20160118 074610\n'),
 ('128.134.45.24', '/gallery/memory.html', '20160118 074611\n'),
 ('128.134.45.24', '/gallery/memory.html', '20160118 074612\n')]
```

groupby를 이용하면 쉽게 분류 가능하다. groupby의 두 번째 인수는 그룹 키를 추출하는 함수이다. 따라서 ip를 중심으로 그룹핑 할경우는 lambda e: e[0]가 된다.

In [16]:
```
from itertools import groupby

data = [tuple(line.split(':')) for line in open('Weblog')]
for key, group in groupby(data, lambda e: e[0]):
    print key, len(list(group))
```

```
128.134.45.23 10
128.134.45.24 5
```

10

In [17]:
```
import sha

uid = 'hongildong'
pwd = 'heogyun'

encrypted = sha.new(pwd).hexdigest()
```

In [18]:
```
encrypted
```

Out[18]:
```
'e9de6b78d9e9dcf984cc166f58d83737c43801ef'
```

In [19]:
```
with open('passwd', 'a') as f:
    f.write('{}:{}\n'.format(uid, encrypted))
```

11

In [20]:
```
def check_login(uid, pwd):
    encrypted = sha.new(pwd).hexdigest()
    for line in open('passwd'):
        uid2, enc2 = line.strip().split(':')
        if uid == uid2 and encrypted == enc2:
            return True
    return False

def login(uid, pwd):
    if check_login(uid, pwd):
        print 'You are a registered member'
    else:
        print 'Sorry, you are not a registered member.'
```

In [21]:
```
uid = 'hongildong'
pwd = 'heogyun'

login(uid, pwd)
```

```
You are a registered member
```

In [22]:
```
uid = 'hongildong2222'
pwd = 'heogyun'

login(uid, pwd)
```

```
Sorry, you are not a registered member.
```

12

In [23]:
```
import urllib
import urlparse
import os
```

In [24]:
```
url = 'https://docs.python.org/2/library/urlparse.html'
p = urlparse.urlparse(url)
p
```

Out[24]:
```
ParseResult(scheme='https', netloc='docs.python.org', path='/2/li
brary/urlparse.html', params='', query='', fragment='')
```

In [25]:
```
def save_html_page(url):
    p = urlparse.urlparse(url)

    html = urllib.urlopen(url).read()
    fpath_to_save = p.path.lstrip('/')
    folder, fname = os.path.split(fpath_to_save)
    if not os.path.exists(folder):
        os.makedirs(folder)
    open(fpath_to_save, 'w').write(html)
    return html
```

In [26]:
```
start_url = 'https://docs.python.org/2/library/urlparse.html'
start_server = urlparse.urlparse(start_url).netloc
html = save_html_page(start_url)
```

HTML 문서를 분석하는 것은 BeautifulSoup을 이용하면 편리하다. BeautifulSoup의 설치는 콘솔에서 다음 명령을 입력한다.

```
pip install beautifulsoup4
```

In [27]:
```
from bs4 import BeautifulSoup

soup = BeautifulSoup(html)
```

In [28]:
```
for tag in soup.select('a'):  # 모든 a 태그에 대해서
    url2 = urlparse.urljoin(url, tag.get('href'))  # href 속성에 따라 url을 만든다
    # 같은 서버이고 .html로 끝나는 url만 필터링한다
    if start_server == urlparse.urlparse(url2).netloc and url2.endswith('.html'):
        print url2
#         save_html_page(url2)
```

```
https://docs.python.org/2/genindex.html
https://docs.python.org/2/py-modindex.html
https://docs.python.org/2/library/socketserver.html
https://docs.python.org/2/library/uuid.html
https://docs.python.org/2/index.html
https://docs.python.org/2/library/index.html
https://docs.python.org/2/library/internet.html
https://docs.python.org/2/contents.html
https://docs.python.org/2/library/urlparse.html
https://docs.python.org/2/library/uuid.html
https://docs.python.org/2/library/socketserver.html
https://docs.python.org/2/bugs.html
https://docs.python.org/2/genindex.html
https://docs.python.org/2/py-modindex.html
https://docs.python.org/2/library/socketserver.html
https://docs.python.org/2/library/uuid.html
https://docs.python.org/2/index.html
https://docs.python.org/2/library/index.html
https://docs.python.org/2/library/internet.html
https://docs.python.org/2/copyright.html
https://docs.python.org/2/bugs.html
```

In [29]:
```

```