## 예제1] part02/01read_csv.py

```python
# -*- coding: utf-8 -*-
import pandas as pd

file_path = './data/read_csv_sample.csv'

df1 = pd.read_csv(file_path)
print(df1)

df2 = pd.read_csv(file_path, header=None)
print(df2)

df3 = pd.read_csv(file_path, index_col=None)
print(df3)

df4 = pd.read_csv(file_path, index_col='c0')
print(df4)

df5 = pd.read_csv(file_path, names=['손오공','저팔계','사오정'])
print(df5)

df6 = pd.read_csv(file_path, skiprows=2)
print(df6)
```

## 예제2] part02/02read_excel_json.py

```python
# -*- coding: utf-8 -*-
import pandas as pd

df1 = pd.read_excel('./data/남북한발전전력량.xlsx', engine='openpyxl')
print(df1)

df2 = pd.read_excel('./data/남북한발전전력량.xlsx', engine='openpyxl',
                    header=None)
print(df2)

df3 = pd.read_excel('./data/남북한발전전력량.xlsx', engine='openpyxl',
                    header=1)
print(df3)


df3 = pd.read_json('./data/read_json_sample.json')
print(df3)
print(df3.index)
```

## 예제3] part02/03read_html.py

```python
# -*- coding: utf-8 -*-
import pandas as pd

url ='./data/sample.html'

tables = pd.read_html(url)

print(len(tables))

for i in range(len(tables)):
    print("tables[%s]" % i)
    print(tables[i])
    print('\n')

df = tables[1]

df.set_index(['name'], inplace=True)
print(df)
```

## 예제4] part02/04read_webscraping.py

```
# -*- coding: utf-8 -*-

from bs4 import BeautifulSoup
import requests
import pandas as pd

url = 'https://kin.naver.com/search/list.nhn?query=%ED%8C%8C%EC%9D%B4%EC%8D%AC'
response = requests.get(url)

dicts = {}

if response.status_code==200:
    html = response.text
    soup = BeautifulSoup(html, 'html.parser')

    title01 = soup.select_one('#s_content > div.section > ul > li:nth-child(1)
    #print("첫번째제목(HTML코드포함):", title01)

    text01 = title01.get_text()
    #print("첫번째제목(텍스트만추출):", text01)

    ul = soup.select_one('#s_content > div.section > ul')
    #print(ul)
    title02 = ul.select('li > dl > dt > a')
    #print(title02)


    cnt = 1
    for tit in title02:
        #print(tit.get_text())
        my_key = '항목' + str(cnt)
        dicts[my_key] = [tit.get_text(), '2행데이터'];
        cnt += 1
    print(dicts)
else:
    print(response.status_code)

df = pd.DataFrame(dicts)
print(df)
```

### 코드복사

| | |
|---|---|
| 1 | https://kin.naver.com/search/list.nhn?query=%ED%8C%8C%EC%9D%B4%EC%8D%AC |
| 2 | #s_content > div.section > ul > li:nth-child(1) > dl > dt |

## 예제5] part02/05to_csv.py

```python
# -*- coding: utf-8 -*-

import pandas as pd

data = {'name' : [ 'Jerry', 'Riah', 'Paul'],
        'algol' : [ "A", "A+", "B"],
        'basic' : [ "C", "B", "B+"],
        'c++' : [ "B+", "C", "C+"],
        }

df = pd.DataFrame(data)
df.set_index('name', inplace=True)

print(df)

df.to_csv("./save/df_sample2.csv")

```

## 예제6] part02/06to_excel.py

```python
# -*- coding: utf-8 -*-

import pandas as pd

data = {'name' : [ 'Jerry', 'Riah', 'Paul'],
        'algol' : [ "A", "A+", "B"],
        'basic' : [ "C", "B", "B+"],
        'c++' : [ "B+", "C", "C+"],
        }

df = pd.DataFrame(data)
df.set_index('name', inplace=True)
print(df)

df.to_excel("./save/df_sample.xlsx")

```

## 예제7] part02/07read_api_to_excel.py

[JSP예제 먼저 해보기]

```
# -*- coding: utf-8 -*-

import os
import sys
import urllib.request
import json
import pandas as pd

client_id = "G8G5oHazpv0RS8kVCWmJ"
client_secret = "bosKhPUu_x"
encText = urllib.parse.quote("JSP책")
url = "https://openapi.naver.com/v1/search/blog?query=" + encText # json 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 
request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id",client_id)
request.add_header("X-Naver-Client-Secret",client_secret)
response = urllib.request.urlopen(request)
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    result = response_body.decode('utf-8')
else:
    print("Error Code:" + rescode)

```

```
title = []
bloggername = []
bloggerlink = []
json_data = json.loads(result)
for item in json_data['items']:
    #print(item['title'])
    title.append(item['title'])
    bloggername.append(item['bloggername'])
    bloggerlink.append(item['bloggerlink'])

df = pd.DataFrame({'제목':title, '작성자':bloggername, '링크':bloggerlink})
print(df)

df.set_index('제목', inplace=True)
df.to_excel("./save/JSP검색결과.xlsx")

```

## 예제8] part02/08excel_writer.py

```python
# -*- coding: utf-8 -*-

import pandas as pd

data1 = {'name' : [ 'Jerry', 'Riah', 'Paul'],
         'algol' : [ "A", "A+", "B"],
         'basic' : [ "C", "B", "B+"],
          'c++' : [ "B+", "C", "C+"]}

data2 = {'c0':[1,2,3],
         'c1':[4,5,6],
         'c2':[7,8,9],
         'c3':[10,11,12],
         'c4':[13,14,15]}

df1 = pd.DataFrame(data1)
df1.set_index('name', inplace=True)
print(df1)
print('\n')

df2 = pd.DataFrame(data2)
df2.set_index('c0', inplace=True)
print(df2)

writer = pd.ExcelWriter("./save/df_excelwriter.xlsx")
df1.to_excel(writer, sheet_name="sheet1")
df2.to_excel(writer, sheet_name="sheet2")
writer.save()

```

# 셀레니움으로 웹크롤링 하기

셀레니움 교안 [바로가기]

## 예제9] 09selenium1_melon.py

url작성시 http ⇒ https 로 변경해주세요.

```
1  # -*- coding: utf-8 -*-
2  from selenium import webdriver
3  driver = webdriver.Chrome('./data/chromedriver.exe')
4
5  url = 'http://www.melon.com/chart/index.htm'
6  driver.get(url)
7  html = driver.page_source
8
9  from bs4 import BeautifulSoup
10 soup = BeautifulSoup(html, 'html.parser')
11
12 song_data = []
13 rank = 1
14 songs = soup.select('tbody > tr')
15 for song in songs:
16     title = song.select('div.ellipsis.rank01 > span > a')[0].text
17     singer = song.select('div.ellipsis.rank02 > a')[0].text
18     print(title, singer, sep="¦")
19     song_data.append(['Melon', rank, title, singer])
20     rank += 1
21
22 import pandas as pd
23 columns = ['서비스','순위','타이틀','가수']
24 pd_data = pd.DataFrame(song_data, columns=columns)
25 print(pd_data.head())
26 pd_data.to_excel('./save/melon.xlsx', index=False)
27
```

## 예제10] 09selenium2_bugs.py

실습 : 1~100위까지 순위를 크롤링해서 엑셀로 저장하시오.

파일명 : bugs.xlsx

벅스챠트URL : https://music.bugs.co.kr/chart

해답 소스 [바로가기]

## 예제11] 09selenium3_naver_login.py

```python
# -*- coding: utf-8 -*-
from selenium import webdriver

driver = webdriver.Chrome('./data/chromedriver.exe')
driver.implicitly_wait(3)
url = 'https://nid.naver.com/nidlogin.login'
driver.get(url)

driver.find_element_by_name('id').send_keys('nak*********')
driver.find_element_by_name('pw').send_keys('본인의비밀번호')

driver.implicitly_wait(5)
driver.find_element_by_xpath('//*[@id="log.login"]').click()

```

## 예제12] 09selenium4_genie.py

```python
# -*- coding: utf-8 -*-

import pandas as pd
from selenium import webdriver
from bs4 import BeautifulSoup

driver = webdriver.Chrome('./data/chromedriver.exe')
url = 'https://www.genie.co.kr/chart/top200'
driver.get(url)

html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

song_data = []
rank = 1
songs = soup.select('tbody > tr')
for song in songs:
    title = song.select('a.title')[0].text.strip()
    singer = song.select('a.artist')[0].text
    print(title, singer, sep="¦")
    song_data.append(['Genie', rank, title, singer])
    rank += 1

##두번째 페이지
driver.implicitly_wait(2) # 암묵적으로 웹 자원을 (최대) 3초 기다리기
driver.find_element_by_xpath(
    '//*[@id="body-content"]/div[7]/a[2]'
).click()
```

…페이지 별로 반복하세요.

```python
##네번째 페이지
driver.implicitly_wait(2) # 암묵적으로 웹 자원을 (최대) 3초 기다리기
driver.find_element_by_xpath(
    '//*[@id="body-content"]/div[7]/a[4]'
).click()

songs = soup.select('tbody > tr')
for song in songs:
    title = song.select('a.title')[0].text.strip()
    singer = song.select('a.artist')[0].text
    print(title, singer, sep="¦")
    song_data.append(['Genie', rank, title, singer])
    rank += 1

columns = ['서비스','순위','타이틀','가수']
pd_data = pd.DataFrame(song_data, columns=columns)
print(pd_data.head())
pd_data.to_excel('./save/genie.xlsx', index=False)
```

```
# -*- coding: utf-8 -*-
#라이브러리 임포트
import pandas as pd
from selenium import webdriver
from bs4 import BeautifulSoup
from selenium.webdriver.common.by import By

#셀레니움 드라이버 로드 및 크롬 브라우저 열기
driver = webdriver.Chrome()
url = 'https://www.genie.co.kr/chart/top200'
driver.get(url)

#페이지 정보를 얻어온후 숩 객체 생성
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

#챠트의 1~50위까지의 데이터를 파싱한 후 리스트에 저장한다.
song_data = []
rank = 1
#리스트(목록)는 대부분 table태그로 제작되므로 tr을 찾은후 반복하면된다.
songs = soup.select('tbody > tr')
for song in songs:
    #제목과 가수를 가져온다.
    title = song.select('a.title')[0].text.strip()
    singer = song.select('a.artist')[0].text
    print(title, singer, sep="|")
    #정보를 리스트에 추가한다.
    song_data.append(['Genie', rank, title, singer])
    rank += 1
#묵시적으로 2초간 대기한다.
driver.implicitly_wait(2)
#xpath를 통해 버튼을 찾은 후 클릭한다.
driver.find_element(By.XPATH,'//*[@id="body-content"]/div[7]/a[2]').click()

#2페이지에 대한 정보를 새로 얻어온 후 크롤링을 시작한다.
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')
songs = soup.select('tbody > tr')
```

```
for song in songs:
    title = song.select('a.title')[0].text.strip()
    singer = song.select('a.artist')[0].text
    print(title, singer, sep="|")
    song_data.append(['Genie', rank, title, singer])
    rank += 1
driver.implicitly_wait(2)
driver.find_element(By.XPATH,
    '//*[@id="body-content"]/div[7]/a[3]'
).click()

#3페이지
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')
songs = soup.select('tbody > tr')
for song in songs:
    title = song.select('a.title')[0].text.strip()
    singer = song.select('a.artist')[0].text
    print(title, singer, sep="|")
    song_data.append(['Genie', rank, title, singer])
    rank += 1
driver.implicitly_wait(2)
driver.find_element(By.XPATH,
    '//*[@id="body-content"]/div[7]/a[4]'
).click()

#4페이지
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')
songs = soup.select('tbody > tr')
for song in songs:
    title = song.select('a.title')[0].text.strip()
    singer = song.select('a.artist')[0].text
    print(title, singer, sep="|")
    song_data.append(['Genie', rank, title, singer])
    rank += 1

#데이터프레임에 컬럼을 추가한 후 변환한다.
columns = ['서비스','순위','타이틀','가수']
```

```
pd_data = pd.DataFrame(song_data, columns=columns)
print(pd_data.head())
#엑셀로 저장한다.
pd_data.to_excel('./save/genie.xlsx', index=False)
```

## 예제13] 10oracle_connect.py

### cx_Oracle 라이브러리 설치가 필요합니다.

윈도우 C++ 설치

```
(base) C:\Users\nakja> conda install cx_Oracle
```

```python
# -*- coding: utf-8 -*-
import cx_Oracle as cx

host_name = 'localhost'
oracle_port = 1521
service_name = 'xe'
connect_info = cx.makedsn(host_name, oracle_port, service_name)
conn = cx.connect('musthave', '1234', connect_info)
cursor = conn.cursor()

# 인파라미터가 없는 쿼리문
sql = "select * from member"
cursor.execute(sql)
print("전체회원출력")
for rs in cursor:
    print(rs[0], rs[1], rs[2], rs[3])

# 인파라미터가 있는 쿼리문
sql = "select * from member where id=:userid"
cursor.execute(sql, userid='test1')
member = cursor.fetchone()
print("\ntest1 회원출력")
print("%s %s %s %s" % (member[0], member[1], member[2], member[3]))

```

```python
# 인서트
my_tit = "제목입니다"
my_con = "내용입니다"
my_id = "musthave"
sql = """insert into board (num,title,content,id,postdate,visitcount)
        values (seq_board_num.nextval, :title, :content, :userid,
                sysdate, 0)"""
try:
    cursor.execute(sql, title=my_tit, content=my_con, userid=my_id)
    conn.commit()
    print("1개의 레코드 입력")
except Exception as e:
    conn.rollback()
    print("insert 실행시 오류발생", e)
conn.close()
```