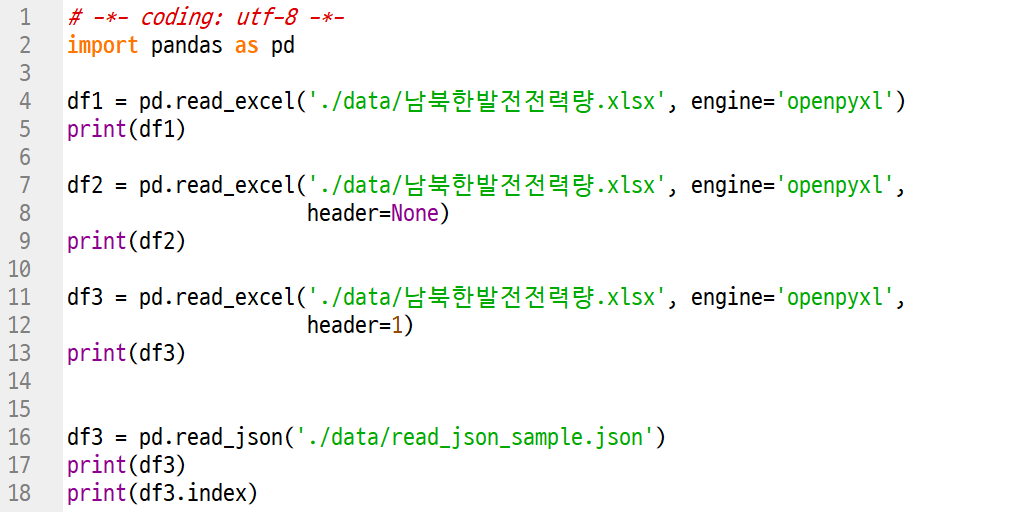
# 예제1] part02/01read\_csv.py



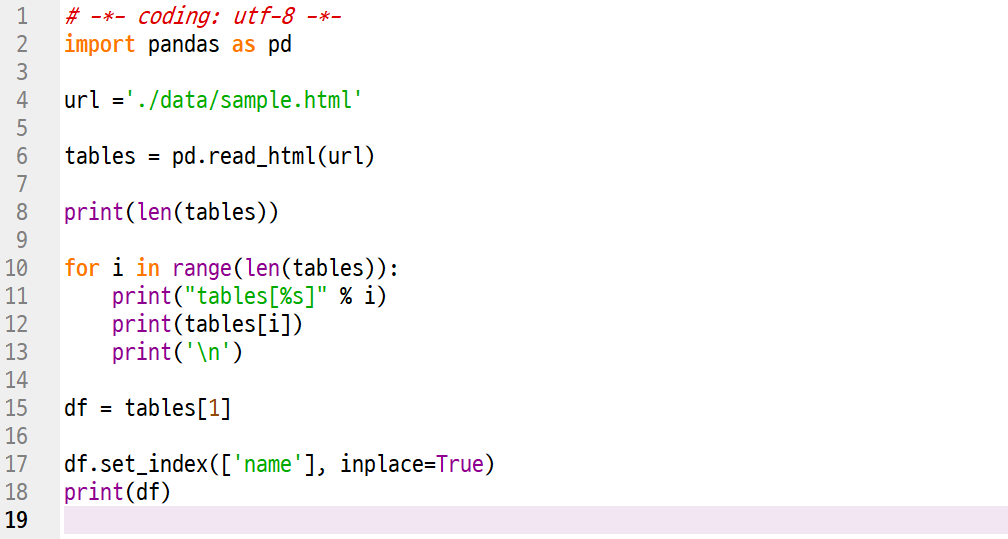
# 

# 예제2] part02/02read\_excel\_json.py



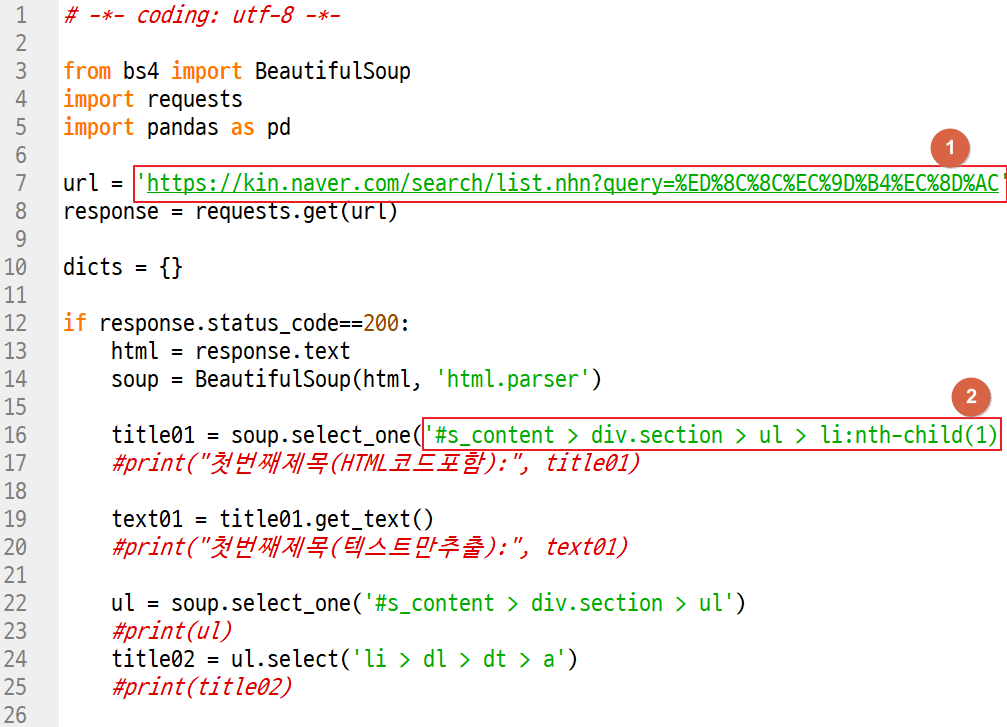
# 

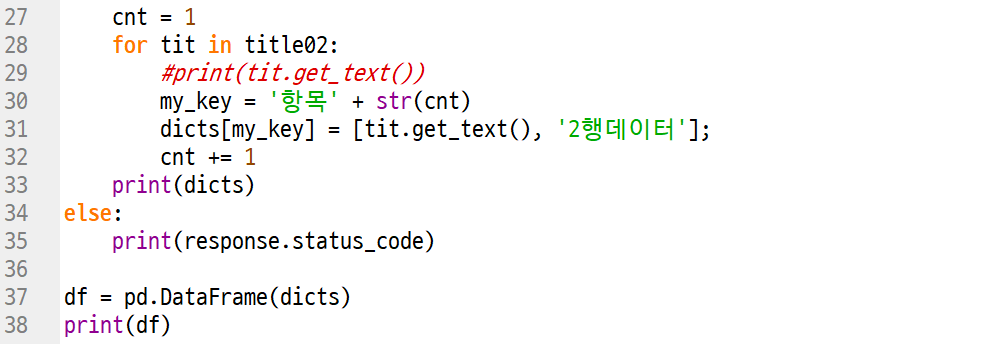
# 예제3] part02/03read\_html.py



# 

# 예제4] part02/04read\_webscraping.py





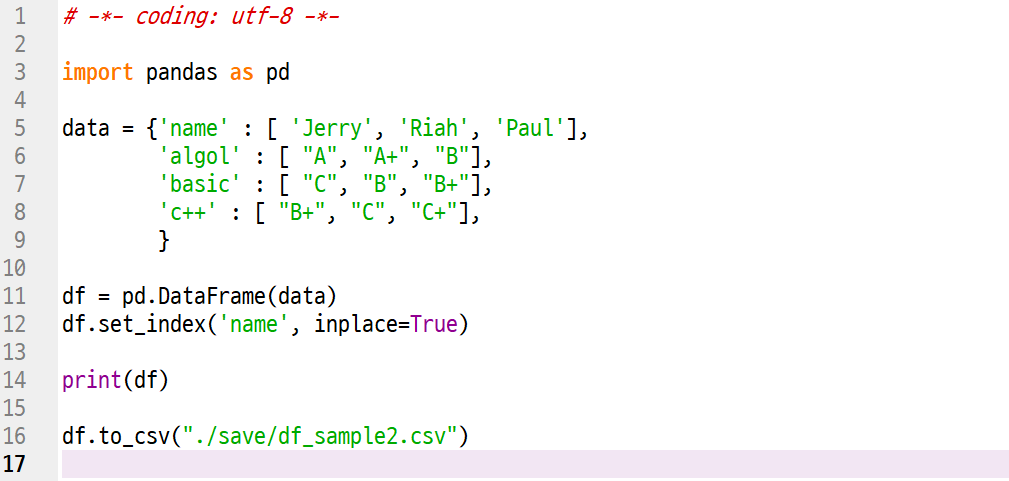
### 코드복사

| 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 |

## 

# 

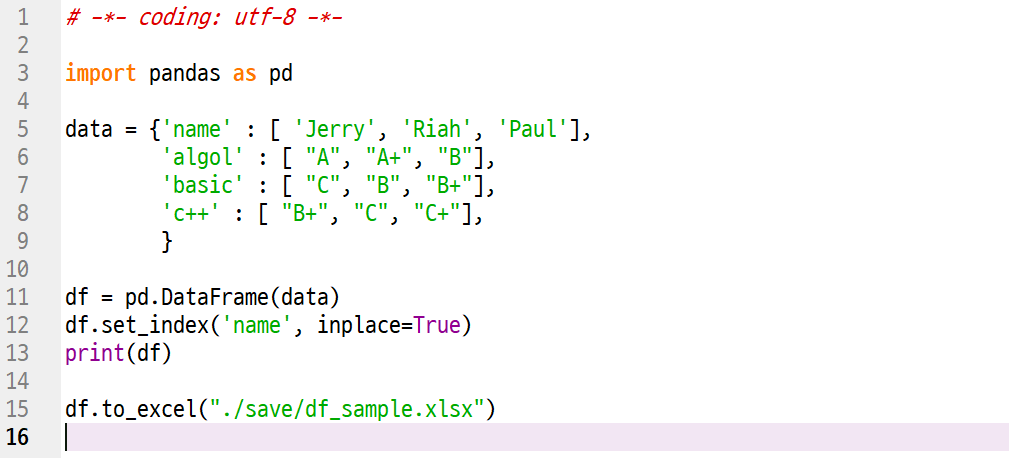
# 예제5] part02/05to\_csv.py



# 

# 

# 예제6] part02/06to\_excel.py



# 

# 예제7] part02/07read\_api\_to\_excel.py

## [[JSP예제 먼저 해보기](https://docs.google.com/document/d/1mjoew3ZGuh7GzCdKHUaVR2tNlGc5IP2C0Wys18aGG4g/edit?usp=drive_link)]

# 

# 

# 예제8] part02/08excel\_writer.py

# 

# 

셀레니움으로 웹크롤링 하기

### 셀레니움 교안 [[바로가기](https://docs.google.com/document/d/1giSXGhPmGWyECmiXM3MQ3VGcgtIyKvWMyyaGOkhMaWI/edit?usp=drive_link)]

# 예제9] 09selenium1\_melon.py

url작성시 http ⇒ https 로 변경해주세요.



# 

# 예제10] 09selenium2\_bugs.py

실습 : 1~100위까지 순위를 크롤링해서 엑셀로 저장하시오.

파일명 : bugs.xlsx

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

해답 소스 [[바로가기](https://docs.google.com/document/d/1Qz9oZZhJ7AVVFD8tTiEKxEc20oYDyevHBBSsG-Xyrgg/edit?usp=sharing)]

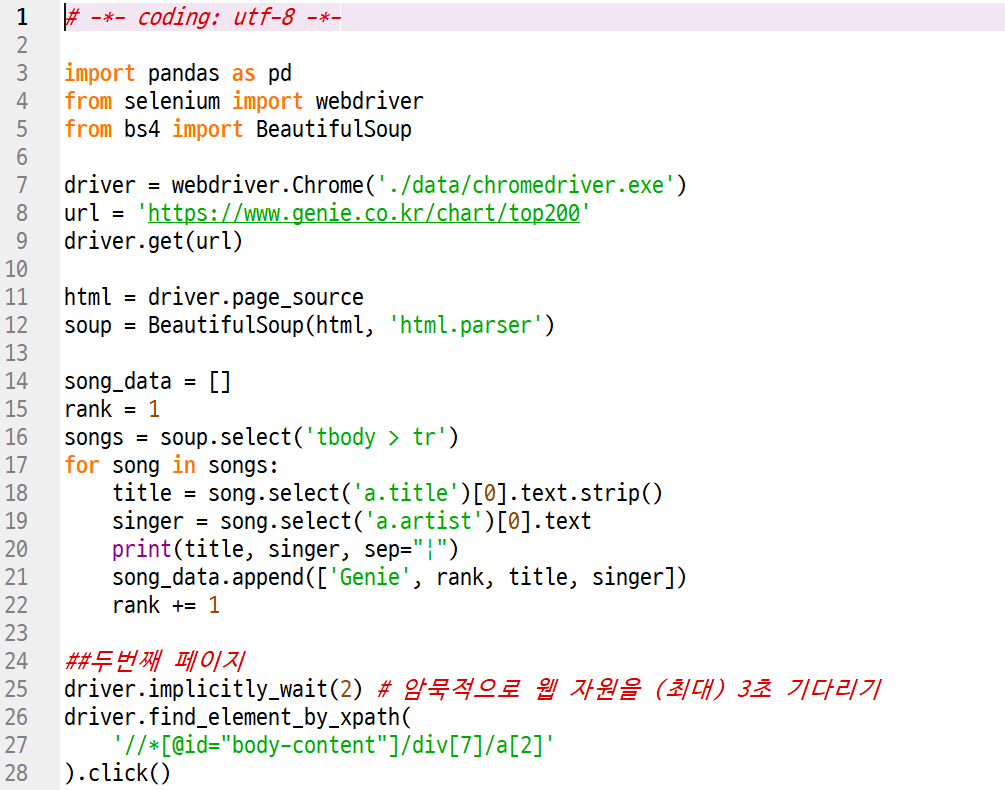
# 

# 예제11] 09selenium3\_naver\_login.py

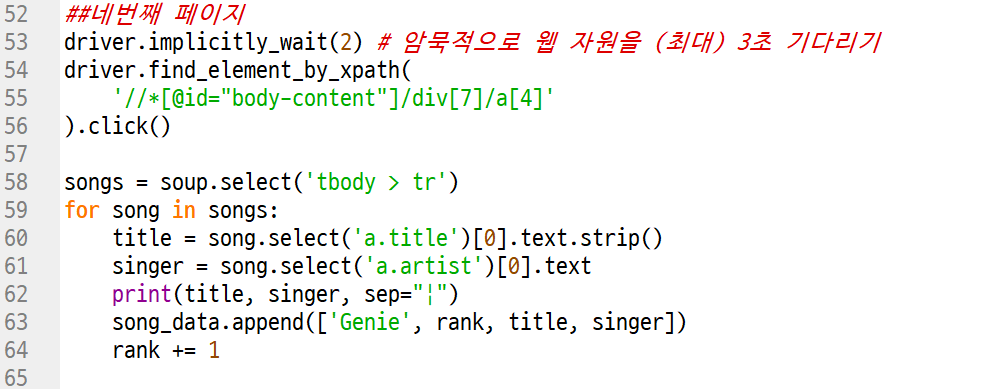


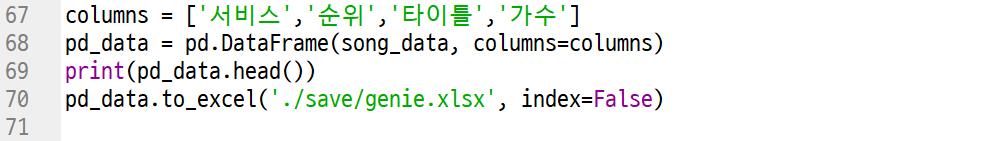
# 

# 예제12] 09selenium4\_genie.py



…페이지 별로 반복하세요.





| # -\*- 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](https://drive.google.com/file/d/16-I9x79VJDySlblqgulfZD4DCdkGqm7V/view?usp=drive_link)

|  | (base) C:\Users\nakja> conda install cx\_Oracle | |
| --- | --- | --- |

