

저는 처음에 click으로 url을 시켜주려고 했으나 더욱 더 편의성을 위해서 url로 직접 들어가서 그 페이지의 자료들을 크롤링하는 것으로 결정을 했습니다.

```
from selenium.webdriver import Chrome
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup
import time
import re
import requests
import csv

chrome = Chrome()

# url로 이동하기
suwon_list=[1,7,13,24,30,34,41,48,52,59,63,70,77,84,87,91,99,104,113,119,125,130,137,140,150,154,160,167,169,177,184,187,193,203,210,214,218,224,230,236,242,248,254,260,266,272,278,284,290,296,302,308,314,320,326,332,338,344,350,356,362,368,374,380]
rounder = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38]
def url_move(suwon_list):
    time.sleep(1)
    url = f"https://www.kleague.com/match.do?year=2022&leagueId=1&gameId={suwon_list}&meetSeq=1&startTabNum=1" # 수원FC
    chrome.get(url)
# 점수(스코어)
def home_score():
    score = chrome.find_element(by='xpath', value='//*[@id="matchInfoArea"]')
    return score.text[0:1]
def away_score():
    score = chrome.find_element(by='xpath', value='//*[@id="matchInfoArea"]')
    return score.text[2:3]
```

우선 수원FC가 경기를 했던 자료들을 뽑기위해서 url을 분석을 해본 결과는 gamid가 다르게 나왔기 때문에 수원FC경기 번호들을 전부 suwon_list에 담아서 경로를 지정해주었고 그것을 만든 함수가 url_move입니다 home_score와away_score도 마찬가지로 home팀의 점수를 나타내고 슬라이싱을 해서 [0:1]을해서 왼쪽 점수를 출력, away의 점수들을 경기마다 출력을 해주기 위해서 슬라이싱을 하여 오른쪽 [2:3]을 꺼내서 text로 출력하였습니다

```
# home팀 주요기록
def home_record():
    li=[]
    for i in range(1,11):
        myform = chrome.find_element(by='xpath', value=f'//*[@id="matchRecordArea"]/li[{i}]/div[2]')
        li.append(myform.text)
    return li

# away팀 주요기록
def away_record():
    li2=[]
    for i in range(1,11):
        myform = chrome.find_element(by='xpath', value=f'//*[@id="matchRecordArea"]/li[{i}]/div[4]')
        li2.append(myform.text)
    return li2

# 현재 평점 출력
def avg(myforms):
    result = 0
    result2 = 0
    for i in range(len(myforms)):
        if i <= 11:
            result += float(myforms[i][0:3])
        else:
            result2 +=float(myforms[i][0:3])
    avg = result/11
    avg2 = result2/11
    return round(avg,2),round(avg2,2)

def myformAvg():
    btn = chrome.find_element(by = 'xpath', value = '/html/body/div[2]/div[ul/li[2]/span').click()
    myform = BeautifulSoup(response).find('div', {'class':'formation'})
    myforms = myform.get_text().replace(' ','').split()
    return avg(myforms) #팀평균

# 홈,아웨이팀
def teamname():
    teamform = BeautifulSoup(response).find('div', {'class':'score-box'})
    team = teamform.select('body > div.sub-top.match > div > div.score-box > div:nth-child(1)')[0].get_text().replace('\n','').replace(' ','')
    team2 = teamform.select('body > div.sub-top.match > div > div.score-box > div:nth-child(3)')[0].get_text().replace('\n','').replace(' ','')
    return team,team2
```

각 팀의 주요기록을 출력해주기 위해서 home팀의 주요기록 : home_record, away팀의 주요기록 : away_record를 만들어주었고 li{}가 규칙적으로 숫자가 바뀌기 때문에 for문을 돌려,

10개의 기록들을 수집을 해주었습니다. 전체 평점 출력은 팀 2개를 해주기위해서 result(home팀),result2(away)로 구분하여 전체 주요기록을 가져와서 앞부분과 뒷부분을 잘라주어서 앞부분은 home, 뒷부분은 away로 구분해줍니다 그리고 정리를 위해서 myformAvg함수를 생성하여 split을 활용하여 잘라줍니다 그리고 그 값을 return , teamname은 각각의 팀 구분을 해줍니다

```
# csv 저장하기
csvFile = open('test.csv', 'a', newline='')
writer = csv.writer(csvFile)
# writer.writerow(['홈', '어웨이', '라운드', '점유율', '슈팅', '유효슈팅', '파울', '경고', '퇴장', '퇴장(경고누적)', '코너킥', '프리킥', '오프사이드'])
# csv파일 저장 함수
def test_csv(writer,rounder):
    #주요기록
    hr1=home_record()
    ar1=away_record()
    hs1=home_score()
    as1=away_score()
    try:
        writer.writerow((
            teamname()[0],#홈
            teamname()[1],#어웨이
            rounder,#라운드
            f'{hr1[0]}:{ar1[0]}',
            f'{hr1[1]}:{ar1[1]}',
            f'{hr1[2]}:{ar1[2]}',
            f'{hr1[3]}:{ar1[3]}',
            f'{hr1[4]}:{ar1[4]}',
            f'{hr1[5]}:{ar1[5]}',
            f'{hr1[6]}:{ar1[6]}',
            f'{hr1[7]}:{ar1[7]}',
            f'{hr1[8]}:{ar1[8]}',
            f'{hr1[9]}:{ar1[9]}',
            f'{myformAvg()[0]}:{myformAvg()[1]}',#평점
            f'{hs1}:{as1}'#스코어
        ))
    finally:
        csvFile.close()
```

csv저장파일을 함수로 정의를 해서 위에있는 함수들을 호출하여 변수로 이름을 만들어 준 후에 writer.writerow를 하여 위에있는 분석을 해준 것들을 순서대로 적어서 csv파일에 저장을 해주고 마지막에는 닫아줍니다.

```
# # 2월 1경기 전북 수원FC 1-0
url_move(suwon_list[0])
test_csv(writer,rounder[0])
```

이런식으로 리스트를 순서대로 설정을 해주고 경기마다 하나씩올려주면서 함수를 사용해주면 편하게 csv파일에 글을 쓰고 저장을 해줄 수 있습니다.