```
# D:\Dropbox\Doug Eagan\Weight\soccer players scraper.py
import pandas as pd
import re
import requests
from bs4 import BeautifulSoup

# Get basic players information for all players
base_url = "https://sofifa.com/players?offset="
columns = ['ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag', 'Overall',
'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Special']
data = pd.DataFrame(columns = columns)

for offset in range(0, 290):
    url = base_url + str(offset * 61)
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text, 'html.parser')
    table_body = soup.find('tbody')
    for row in table_body.findAll('tr'):
        td = row.findAll('td')
        picture = td[0].find('img').get('data-src')
        pid = td[0].find('img').get('id')
        nationality = td[1].find('a').get('title')
        flag_img = td[1].find('img').get('data-src')
        name = td[1].findAll('a')[1].text
        age = td[2].find('div').text.strip()
        overall = td[3].text.strip()
        potential = td[4].text.strip()
        club = td[5].find('a').text
        club_logo = td[5].find('img').get('data-src')
        value = td[6].text.strip()
        wage = td[7].text.strip()
        special = td[8].text.strip()
        player_data = pd.DataFrame([[pid, name, age, picture, nationality,
flag_img, overall, potential, club, club_logo, value, wage, special]])
        player_data.columns = columns
        data = data.append(player_data, ignore_index=True)
data = data.drop_duplicates()

# Get detailed player information from player page
detailed_columns = ['Preferred Foot', 'International Reputation', 'Weak Foot',
'Skill Moves', 'Work Rate', 'Body Type', 'Real Face', 'Position', 'Jersey Number',
'Joined', 'Loaned From', 'Contract Valid Until','Birthdate', 'Height', 'Weight',
'LS', 'ST', 'RS', 'LW', 'LF', 'CF', 'RF', 'RW', 'LAM', 'CAM', 'RAM', 'LM', 'LCM',
'CM', 'RCM', 'RM', 'LWB', 'LDM', 'CDM', 'RDM', 'RWB', 'LB', 'LCB', 'CB', 'RCB', 'RB',
'Crossing', 'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribbling',
'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration', 'SprintSpeed',
'Agility', 'Reactions', 'Balance', 'ShotPower', 'Jumping', 'Stamina', 'Strength',
'LongShots', 'Aggression', 'Interceptions', 'Positioning', 'Vision', 'Penalties',
'Composure', 'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
'GKKicking', 'GKPositioning', 'GKReflexes', 'ID']
detailed_data = pd.DataFrame(index = range(0, data.count()[0]), columns =
detailed_columns)
detailed_data.ID = data.ID.values

player_data_url = 'https://sofifa.com/player/'
for id in data.ID:
    url = player_data_url + str(id)
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text, 'html.parser')
    skill_map = {}
    columns = soup.find('div', {'class': 'teams'}).find('div', {'class':
'columns'}).findAll('div', {'class': 'column col-4'})
    for column in columns:
        skills = column.findAll('li')
        for skill in skills:
```

```
            if(skill.find('label') != None):
                label = skill.find('label').text
                value = skill.text.replace(label, '').strip()
                skill_map[label] = value
    meta_data = soup.find('div', {'class': 'meta'}).text.split(' ')
    length = len(meta_data)
    birthyear = meta_data[length - 3]
    birthday = meta_data[length - 4]
    birthmonth = meta_data[length - 5]
    weight = meta_data[length - 1]
    height = meta_data[length - 2].split('\'')[0] + '\'' + meta_data[length -
2].split('\'')[1].split('\"')[0]
    skill_map["Birthdate"] = birthmonth+"/"+birthday+"/"+birthyear
    skill_map["Height"] = height
    skill_map['Weight'] = weight
    if('Position' in skill_map.keys()):
        if skill_map['Position'] in ('', 'RES', 'SUB'):
            skill_map['Position'] = soup.find('article').find('div', {'class':
'meta'}).find('span').text
        if(skill_map['Position'] != 'GK'):
            card_rows = soup.find('aside').find('div', {'class': 'card
mb-2'}).find('div', {'class': 'card-body'}).findAll('div', {'class': 'columns'})
            for c_row in card_rows:
                attributes = c_row.findAll('div', {'class': re.compile('column
col-sm-2 text-center')})
                for attribute in attributes:
                    if(attribute.find('div')):
                        name = ''.join(re.findall('[a-zA-Z]', attribute.text))
                        value = attribute.text.replace(name, '').strip()
                        skill_map[str(name)] = value
    sections = soup.find('article').findAll('div', {'class': 'mb-2'})[1:3]
    first = sections[0].findAll('div', {'class': 'column col-4'})
    second = sections[1].findAll('div', {'class': 'column col-4'})[:-1]
    sections = first + second
    for section in sections:
        items = section.find('ul').findAll('li')
        for item in items:
            value = int(re.findall(r'\d+', item.text)[0])
            name = ''.join(re.findall('[a-zA-Z]*', item.text))
            skill_map[str(name)] = value
    for key, value in skill_map.items():
        detailed_data.loc[detailed_data.ID == id, key] = value

full_data = pd.merge(data, detailed_data, how = 'inner', on = 'ID')
full_data.to_csv('data.csv', encoding='utf-8-sig')
```