

Bursa Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Python Programlama Proje Ödevi Raporu
MOUSSA BANE 20360859102

Proje amacı: Python Selenium WebDriver kullanılarak Web Scraping yapma işlemi.

https://www.8notes.com/piano/classical/sheet_music/sayfasi üzerinde scraping işlemi yapılacaktır.

1. Listelerdeki tüm parçaların özgün erişim bağlantıları elde edilecek. (Örn: Chopin – Prelude: <https://www.8notes.com/scores/9765.asp>)
2. Bu bağlantılar, elde edildikleri anda birer JSON objesi olarak kodlanacak. Tüm parçalar çekildikten sonra elde edilmiş olan JSON objesi dosyaya yazdırılacak.
 - a. Parçanın porte imgesinin indirme bağlantısı elde edilecek. (Parçanın özgün sayfasından elde edilecek.) “img”
 - b. Parçanın MIDI dosya indirme bağlantısı elde edilecek. (Parçanın özgün sayfasından elde edilecek.) “midi”
 - c. Parçanın hakkında bilgisi elde edilecek. (Parçanın özgün sayfasından elde edilecek.) “about”
 - d. Parçanın zorluk bilgisi elde edilecek. (Tüm parçaların listelendiği sayfadan edinilecek.) “difficulty”

Bu proje gerçekleştirmek için bazı gereken kurulumların yapılması gerekir. Onlardan bahsetmek gerekirse, “pyhon pip ” Python package management için. <https://phoenixnap.com/kb/install-pipwindows#ftoc-heading-1> Bu siteden faydalanarak pip kurulumu yapabilirsiniz.

Ondan sonra, web scraper’imiz için kullanacağımız Python packages : belirli verileri seçmek

için(*BeautifulSoup*) ve dinamik olarak yüklenen içeriği işlemek için(*Selenium*) , indirmek için:

Termalde şu commandları çalıştırmamız gerekir:

pip install beautifulsoup4 ve pip install selenium

Son olarak, makinenize Google Chrome ve Chrome Sürücüsünü yüklediğinizden emin olmaktır. Bunun için <https://www.youtube.com/watch?v=UOsRrxMKJYk> bu videodan faydalanabilirsiniz.

```
1 import re #Regular expression operasyonlari icin
2 import json
3 from bs4 import BeautifulSoup
4 from selenium import webdriver
```

Burada, webten çekeceğimiz verileri işlememizde özel karakterlerden uzak durmak adına “re (regular expression)” import ediyoruz. Ardından çekeceğimiz verileri bir json dosyasında tutacağımız için “json” da import ediyoruz. Son import olarak “BeautifulSoup” ve selenium’den “webdriver” import ediyoruz.

```
7 with open("data.json", "w") as f: #scraped datalarımızı data.json dosyasında tutacağız
8     json.dump([], f)
9
10 def write_json(new_data, filename='data.json'): #Dictionary olarak elimizde olan scraped dataları dosyamıza yazılır
11     with open(filename, 'r+') as file:
12         # İlk başta dosyamızda olan data bir dict'te tutuyoruz
13         file_data = json.load(file)
14         # Sonra yeni gelen dict(scraped data) olan dict'imize append edilir
15         file_data.append(new_data)
16         # Sets file's current position at offset.
17         file.seek(0)
18         # Son olarak tekrardan json'a donusturulur ... indent = 4 ==> json formati icin
19         json.dump(file_data, file, indent = 4)
```

Bu kısımda ise, uygulamamızı çalıştırdığımızda, çektiğimiz dataları tutacağımız json dosya “data.json” adıyla ve “w” moduyla “with” kullanarak açıp kapatarak oluşturuyoruz.

Ardından, sırasıyla bir dictionary(sözlük) titinde ve ilgileneceğimiz dosya adı veya boş bırakıldığında default olarak daha önce oluşturduğumuz “data.json” dosyası parametrelili write_json fonksiyonumuzu oluşturuyoruz. Bu fonksiyonun amacı: ilgili sitelerden çektiğimiz verileri ‘img’ ‘midi’ ‘about’ ve ‘difficulty’ key’leri olan elde ettiğimiz dictionary alıp json dosyamıza yerleştirmektir.

```
23 driver = webdriver.Chrome()
24 page = driver.get('https://www.8notes.com/piano/classical/sheet_music/') # Getting page HTML through request
25 soup = BeautifulSoup(driver.page_source, 'html.parser') # Parsing content using beautifulsoup
26
27 links = soup.select("table tbody tr") # Selecting all the songs
28 difficulties = soup.select("table.table_list tbody tr td.level_type img") ### ==>"difficulty"
29
30 i = 0
31 for difficulty in difficulties: #Tüm parçaların listelendiği sayfadan Parçanın zorluk bilgisi için
32     difficulty = difficulty.get_attribute_list('alt')
33     difficulties[i] = difficulty[0]
34     i += 1
```

Burada ise, tüm parçaların olduğu sayfada ‘difficulty’ keyin value’ları elde ediyoruz. Önce sitenin içeriğin yapısı anlamak için şu https://www.8notes.com/piano/classical/sheet_music/ linke tıkladıktan sonra sağ tıklayıp ‘Öğeyi İncele / Inspect(View Page Source)’ seçeneği seçmeliyiz. “links” değişkeni parçaların listesi tutuyor. “difficulties” ise o sayfadan bütün parçaların difficulty özellikleri tutuyor. For’un amacı ise difficulty bilgileri string olarak güncellemektir.

```

38 j = 0
39 for link in links: #Her parçanın özgün sayfasından data scrape işlemler için
40     link_attribute = link.get_attribute_list('onclick')
41     link_attribute = link_attribute[0][20:-1]
42     #print(link_attribute)
43
44     parcalarinUrl = 'https://www.8notes.com/' + link_attribute
45     #elements_url.append(parcalarinUrl)
46     #print(parcalarinUrl)
47
48     driver.get(parcalarinUrl) #Parçanın özgün sayfasına gidilir
49     newSoup = BeautifulSoup(driver.page_source, 'html.parser')
50     porteInfoList = newSoup.select('ul li a.mp3_list')[0]
51     porteInfoList = porteInfoList.get_attribute_list('href')[0]
52     #print(infoList)
53
54     downloadUrl = 'https://www.8notes.com/' + porteInfoList[1:] ### ==>"img"
55     #print(downloadUrl)
56
57     midiInfoList = newSoup.select('div ul li a.midi_list')[0]
58     midiInfoList = midiInfoList.get_attribute_list('href')[0]
59
60     midiIndirUrl = 'https://www.8notes.com/' + midiInfoList[1:] ### ==>"midi"
61     #print(midiIndirUrl)

```

Bu for'un sayesinde her parçanın tek tek olarak özgün sayfasından ilgili verileri yani 'img' 'midi' ve 'about' keylerin value'ları elde ediyoruz . 'links' tüm parçaların listesi tuttuğu için (for link in links) yaparak her parçanın özgün sayfasına ulaşmamıza olanak sağlar.

İlk sayfada listenin her bir satırı bir linktir, her satırın 'onclick' özelliğinde ilgili özgün sayfanın location'u elde ediyoruz . Daha sonra 'parcalarinUrl' değişkeninde parçanın özgün sayfasının url'i string olarak tutuyoruz. Parçanın sayfasına gitmek için 'driver.get(parcalarinUrl)' fonksiyonu kullandım. Ondan sonra sayfanın kaynak konu pars etmek için yeni bir BeautifulSoup objesi(newSoup) tanımladım. 54.satırında ise 'downloadUrl' değişkeninde parçanın **Porte imgesinin indirme bağlantısı** tutuyoruz. Daha sonra, 'midiIndirUrl' değişkeninde **Midi dosya indirme bağlantısı** tutuyoruz.

```

63 aboutInfoList = newSoup.select("table.comp_table tbody tr td div.artist_col2")
64 if len(aboutInfoList) == 4 : #Bazı sayfaların about kısmı diğerlerden farklı olduğu için
65     aboutDict = {
66         "Title" : newSoup.select('table.comp_table tbody tr td h2')[0].text,
67         "Artist" : re.sub('[^a-zA-Z0-9 \.]', ' ', aboutInfoList[0].text),
68         "Born" : re.sub('[^a-zA-Z0-9 \.]', ' ', aboutInfoList[1].text),
69         "Died" : re.sub('[^a-zA-Z0-9 \.]', ' ', aboutInfoList[2].text) ,
70         "The Artist" : re.sub('[^a-zA-Z0-9 \.]', ' ', aboutInfoList[3].text) ,
71     }
72 else:
73     aboutDict = { #about kısmını diğerlerden farklı olanların link yazılsın
74         "about_url": parcalarinUrl
75     }
76     #print(aboutDict)

```

Bu kısımda ise, 'about' key'in value'yu belirtilecektir. Parçanın özgün sayfasında altta about kısmı mevcuttur. About kısmındaki 'Title', 'Artist', 'Born', 'Died' ve 'The Artist' bilgileri bir dictionary olarak toplayıp 'about' key'in value'u belirtmiş olmaktadır. If-Else kullandığımın sebebi şu, yani bütün sayfaların about kısmının yapısı aynı değildir, parçaların çoğu bu yapıda olduğu için bunu tercih

ettim. Bu yapıda olmayan çarçaların 'about' ise parçanın url olarak belirlenmiş olur. Import ettiğimiz re(regular expression) bu kısımda kullanışı görebiliriz. Örnek bu satırda: `re.sub('[^a-zA-Z0-9\\.]', ' ', aboutInfoList[0].text)`

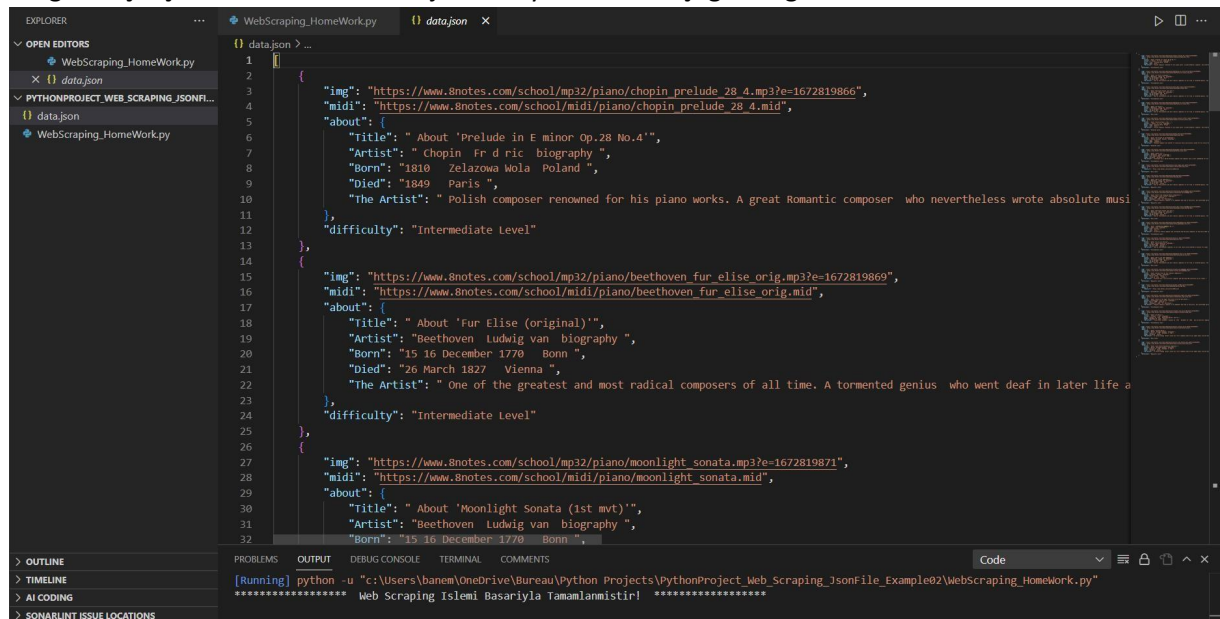
.sub() fonksiyonun ilk parametresi strigimizde kabul edilmek istediğimiz karakterleri belirliyoruz , ikinci parametre ise stringimiz giriyoruz. Dönüş olarak belirlediğimiz karakterlerin haricinde stringimizde başka bir karakter olmayacaktır. Başka karakterlerin yerine bir boşluk yerleştirilir.

```
78     totalScrapedInfo = {
79         "img" : downloadUrl ,                #Parçanın porte imgesinin indirme bağlantısı
80         "midi" : midiIndirUrl ,              #Parçanın MIDI dosya indirme bağlantısı
81         "about" : aboutDict ,                #Parçanın hakkında bilgisi
82         "difficulty" : difficulties[j] ,      #Parçanın zorluk bilgisi
83     }
84     j += 1
85
86
87     #Artık datalarımızı json dosyamıza yazılır
88     write_json(totalScrapedInfo)
89
```

For'un son kısmı olarak ise, elde ettiğimiz verileri bir dictionary aracılığıyla topluyoruz. Ardından json dosyamıza yazdırmak üzere **write_json()** fonksiyonu çağırarak her parça için topladığımız 'img' 'midi' 'about' ve 'difficulty' bilgileri web'ten çekip json dosyaya tutmuş oluruz.

Son olarak, scraping işlemi sorunsuz çalışıp bittiğinde consola haber vermek üzere " Web Scraping Islemi Basariyla Tamamlanmistir " yazdırıyoruz.

Programı çalıştırıldıktan sonra data.json dosyasının hali aşağıdaki gibidir.



```
1  [
2  {
3      "img": "https://www.8notes.com/school/mp32/piano/chopin_prelude_28_4.mp3?e=1672819866",
4      "midi": "https://www.8notes.com/school/midi/piano/chopin_prelude_28_4.mid",
5      "about": {
6          "Title": "About 'Prelude in E minor Op.28 No.4'",
7          "Artist": "Chopin Fr d ric biography ",
8          "Born": "1810  Zelazowa Wola  Poland ",
9          "Died": "1849  Paris ",
10         "The Artist": " Polish composer renowned for his piano works. A great Romantic composer  who nevertheless wrote absolute musi
11     },
12     "difficulty": "Intermediate level"
13 },
14 {
15     "img": "https://www.8notes.com/school/mp32/piano/beethoven_fur_elise_orig.mp3?e=1672819869",
16     "midi": "https://www.8notes.com/school/midi/piano/beethoven_fur_elise_orig.mid",
17     "about": {
18         "Title": "About 'Fur Elise (original)'",
19         "Artist": "Beethoven Ludwig van biography ",
20         "Born": "15 16 December 1770  Bonn ",
21         "Died": "26 March 1827  Vienna ",
22         "The Artist": " One of the greatest and most radical composers of all time. A tormented genius  who went deaf in later life a
23     },
24     "difficulty": "Intermediate level"
25 },
26 {
27     "img": "https://www.8notes.com/school/mp32/piano/moonlight_sonata.mp3?e=1672819871",
28     "midi": "https://www.8notes.com/school/midi/piano/moonlight_sonata.mid",
29     "about": {
30         "Title": "About 'Moonlight Sonata (1st mvt)'",
31         "Artist": "Beethoven Ludwig van biography ",
32         "Born": "15 16 December 1770  Bonn "
```

Proje Kodları İçin Github Reposu:

https://github.com/MoussaBane/PythonProject_WebScraping_With_Python_and_Selenium

Proje Demo Videosu: <https://www.youtube.com/watch?v=mMCQ6BtGS6Q>