LAB-8

Burak uzun 210911131

Not:database işine kafam yatmadığı için database ye kaydedemedim onun dışında bir sorun yok.

1-soru 1:

Özet:

öncelikl get soup fonksiyonundan başlamak istiyorum.bu fonksiyon bize verilen linkteki sitenin html kodlarını doğru bir biçimde karşımıza çıkarmayı sağlar.

Çözüm mimarisi:

Sorunu çözümü için fonksiyona link gönderiyoruz.burda rutin derleme işleminden farklı bir şey yapmıyorum o yüzden kısa kesiyorum.

Çözüm fonksiyonu:

def get\_soup(self,webpage):

        """girdi olarak verilan sayfanın linkindeki html kodlarını güzelleştirip return eder

        Args:

        input\_file\_name: webpage

        output\_file\_name: soup

        Returns:

        soup

        """

        self.gelensayfalinki=webpage

        self.c = urllib2.urlopen(self.gelensayfalinki)

        self.metin = self.c.read()

        self.soup = BeautifulSoup(self.metin,features="html.parser")

        self.soup.prettify()

        return self.soup

2- soru2:

Özet:

Get\_Categories fonksiyonu , bize verilen main fonksiyonun side bar da olan kategorilerini ayrıştırma imkanı sunuyor. Bunları bir sözlüğe hapsedebiliyor.

Çözüm mimarisi:

Öncelikle main linki yani sitenin ana sayfasının linkini mainsite değişkenine tanımladım. Ardından bu site linkini get\_soup fonksiyonuna gönderdim. Bize return edilen soup üzerinden navlistlerin araştırmasını yaptım ve bize a\_listesini verdi. Bu listenin içinde,htmllerinden arıştırılmamış link ve kategoriler var. 42. Satırda bir for döngüsü açtım ve bu listenin içinde htmlleri arındırmak için dödnüm. Bu for döngüsü kısaca,htmllerde arındırıp,çok temiz bir ikili kategori,link listesi veriyor.

Çözüm fonksiyonu:

 def get\_categories(self):

        """main sitedeki kategorileri ve kategorilerin linklerini sözlük olarak dışa verir

        Args:

        input\_file\_name: none

        Returns:

        kategorisözlüğü

        """

        self.mainsite="http://books.toscrape.com/index.html"

        self.soup=self.get\_soup(self.mainsite)

        self.navlistler = self.soup.find\_all(class\_='nav nav-list')

        self.a\_listesi = self.navlistler[0].findChildren("a" , recursive=True)

        self.kategorisozlugu={}

        for i in range(1,len(self.a\_listesi)):

            self.stripcategories=self.a\_listesi[i].string.strip()

            self.link=self.a\_listesi[i].get("href")

            self.link\_birlesmis = urljoin(self.mainsite, self.link)

            self.kategorisozlugu[self.stripcategories]=self.link\_birlesmis

        return self.kategorisozlugu

3-soru 3:

Özet:

Verilen kategorilerin içindeki kitap bilgilerini çeker ve sözlüğe atar.

Çözüm mimarisi:

Fonksiyon argüman alan bir fonksiyon önce soup ve link adında değişkenler alır. Bunlardan soup olan bize,kitap bilgilerini çekebilmemiz için temiz bir çıktı sağlıyor. Link ise kitapların linklerini ilerde join etmek için gerekli.

Öncelikle sitedeki h3 etiketlerini bulum. Bu etiketin içiinde kitap isimleri var. Değer değişkeninde,kitabın fiyatı var.nesneler adında değişken oluşturdum bu nesneyi oluşturmamın sebebi for döngüsünün içinde oluşturduğum sözlüğü doğrudan listeye append etmek. For döngüsünün içine geçelim kitaplar sözlüğü oluşturdum bu oluşumun peşinden değişken isimlerinden anlaşıldığı gibi kitao ismi fiyatı ve href yani link i geliyor. Ardınan bu linki joinliyoruz yani birleştiriyoruz ve kitabın linkini ortaya çıkartıyoruz. Ardınan bunlardan sözlükler sözlüğü oluşturup bir listeye append ediyoruz.

Çözüm fonksiyonu:

def get\_prices\_stars(self,soup,link):

        """verilen kategorilerin içindeki kitapların bilgilerini çeker ve dışarıya atar

        Args:

        input\_file\_name: soup,link

        Returns:

        nesneler

        """

        self.link=link

        self.h3=soup.find\_all("h3")

        self.deger=soup.find\_all(class\_="price\_color")

        self.nesneler=[]

        for i in range(len(self.h3)):

            self.kitaplarSoz={}

            self.kitapAdi=self.h3[i].string.strip().strip()

            self.KitapFiyati=self.deger[i].string

            self.a\_listesi = self.h3[i].findChildren("a" , recursive=True)

            self.a\_linki=self.a\_listesi[0].get("href")

            self.joinlink=urljoin("{}".format(self.link),self.a\_linki)

            self.kitaplarSoz["Name"]=self.kitapAdi

            self.kitaplarSoz["Price"]=self.KitapFiyati

            self.kitaplarSoz["Url"]=self.joinlink

            self.nesneler.append(self.kitaplarSoz)

        return self.nesneler

4-soru 4:

Özet:

Get\_Categories içindeki kategoriler ve linkler içinde for döngüsüyle gezer ve bunları get\_prices\_Star içine gönderir.

Çözüm mimarisi:

Bu işlemin takibini yapabilmek için bir tane proress bar ekledim , ilk 3 değişken progress bar ile ilgili.

107. satırda genel bir sözlük oluşturuyorum bu sözlük bütün kategorilerin bilgilerini barındıracak. For döngüsünün içinde kategorileri geziyor link ve kategorileri temizlemek için get\_soup a gönderiyor. Ardından souplusayfa değişkenini prices\_stars a gönderiyoruz fonksiyondan işlenen veriler genel sözlüke kat=kategori kat1:kitap bilgileri olarak kaydediliyor. Bu satırdan sonraki satır progress bar içindir.

Çözüm fonksiyonu:

 def parse(self):

        """kategorileri tarar ve get\_prices\_stars fonksiyonuna gönderir ve bütün

        kategorideki kitapların bilgilerini bir sözlükte depolar

        Args:

        nıne

        Returns:

        genelsozluk

        """

        total = 50  # total number to reach

        bar\_length = 34  # should be less than 100

        i=0

        self.genelsozluk={}

        self.x=self.get\_categories()

        for kat,lik in self.x.items():

            self.souplusayfa=self.get\_soup(lik)

            self.KAT1=self.get\_prices\_stars(self.souplusayfa,lik)

            self.genelsozluk[kat]=self.KAT1

            ###progress bar

            i=i+1

            percent = 100.0\*i/total

            sys.stdout.write('\r')

            sys.stdout.write("Completed: [{:{}}] {:>3}%"

                            .format('='\*int(percent/(100.0/bar\_length)),

                                    bar\_length, int(percent)))

            sys.stdout.flush()

            time.sleep(0.002)

            ###progress bar

        return self.genelsozluk

Test fonksiyonu:

from BURAK\_UZUN\_211911131\_LAB8 import WebScrapper

def main():

     a=WebScrapper()

     #yorumu kaldırırsak sadece kategori ve link sözlüğü verir.

     #print(a.get\_categories())

    #yorummu kaldırırsak bütün kitap bilgileri ve kategorileri verir.

     #print(a.parse())

main()

test çıktısı:



Soru1 test çıktısı

 Progress bar



2-3-4. soruların ortak çıktısı

Ek dosyalar:

BURAK\_UZUN\_211911131\_LAB8.py

BURAK\_UZUN\_211911131\_LAB8.pdf