



Python BeautifulSoup Modülü

Html veya Xml parse işlemlerinizi python ile yapabilirsiniz.

Tarih: 30-10-2017

BeautifulSoup, HTML veya XML dosyalarını işlemek için oluşturulmuş güçlü ve hızlı bir kütüphanedir. Adını Alice harikalar diyarında içerisindeki bir kaplumbağanın söylediği hikayeden alır.

Bu modül ile bir kaynak içerisindeki HTML kodlarını ayırtıp sadece istediğimiz alanları kesen programlar, daha popüler adıyla BOT yazabilirsiniz.



Kurulum

Ben pip3 ile kurmuştum, pip3 python3 için paket yöneticisi.

```
sudo apt-get install python3-pip
```

komutu ile pip3 kullanmaya başlayabiliriz. Eğer pip3 yüküyse buna gerek yok, şimdi bu modülü kuralım.

```
pip3 install beautifulsoup4
```

Artık modülü kurduk, projemiz içeresine

```
from bs4 import BeautifulSoup
```

diyerek aktaralım. Bu modül ile bir websitesinin HTML kodlarını alacağımız için daha önce detaylıca [anlatığım](#) Requests modülünü de kullanacağız.

```
import requests
```

Artık kodlarımıza yazabiliriz.

Not: Ben örneklerimde <https://www.producthunt.com> sitesine bağlanıp burdaki kodları parse ettim.

Siteye Bağlanma ve Parse İşlemine Hazırlık

```
>>> import requests  
>>> from bs4 import BeautifulSoup  
>>> r = requests.get('https://www.producthunt.com')  
>>> source = BeautifulSoup(r.content, "lxml")  
>>> source.title  
<title>Product Hunt</title>
```

Requests modülü ile siteye bir GET isteği yaptıktı bunu bir r objesine atadık, sonra da BeautifulSoup içine **r.content** değerini verdik, burda hangi kütüphane ile parse etmesini istediğimizi siz belirliyorsunuz. Ben **lxml** seçtim, çünkü hafif ve hızlı, kullanabileceğiniz kütüphaneler şunlar;

- lxml
- html.parser
- lxml-xml
- html5lib

Ardından source objesine sayfanın HTML kodlarını aldı, kontrol etmek için de **source.title** ile sayfanın title etiketi içindeki değerini bastırdık. Herşey başarılı gözüküyor. Bundan sonrası örneklerde yukarıdaki kodları tekrar yazmadan direkt source objesinden ilerleyeceğim.

find(value)

Sayfa kaynağında geçen özel bir değeri çekmenize imkan tanır.

```
>>> source.find("p")  
<p class="text_44214">Product Hunt surfaces the best new products, every day. It's a place for product-lo
```

Bu örnek içerisinde sayfadaki ilk p etiketi içindeki değeri ekrana getirdi. Dikkat ettiyseniz html etiketleri hala duruyor, eğer bunları da kaldırırmak ve sadece yazıya ulaşmak ıstırsınız ise **text** methodunu kullanabilirsiniz.

```
>>> source.find("p").text  
"Product Hunt surfaces the best new products, every day. It's a place for product-lo
```

Başarıyla yazdı! Peki siz sitedeki belirli bir seçiciyi sahip bir url adresini çekmek istiyorsunuz, mesela a class="item_1523a" olan url adresini getirmek istesem?

```
>>> source.find("a", attrs={"class": "item_1523a"}).text  
'Home'
```

Bu bize <https://www.producthunt.com/> sitesindeki solda bulunan FEEDS menüsündeki ilk bağlantı adresinin adını getirdi, yani Home. Bunu almak için de **attrs** sözlüğünü parametre olarak geçtik. Aslında attrs yazmadan direkt olarak sözlüğü de parametre olarak geçebilirsiniz ben yazmayı seviyorum.

find_all(value)

Sayfa kaynağından geçen seçtiğiniz tüm özel değerleri çekmenize imkan tanır. Yani find ile 1 adet, find_all ile tüm değerleri çekiyoruz. Yukarıdaki örneği find_all ile yapalım.

```
>>> solmenu = source.find_all("a", attrs={"class": "item_1523a"})
>>> for link in solmenu:
...     print(link.text)
...
Home
Tech
Games
Books
Artificial Intelligence
Developer Tools
Home
Productivity
Touch Bar Apps
Wearables
All Topics
Customize Your Feed
>>>
```

Gördüğünüz gibi sitedeki sol taraftaki bağlantı adreslerinin hepsini çektiğimiz. Burda dikkat etmeniz gereken şu, find_all ile sayfadaki tüm değerleri çektiğimiz için, bunu bir döngüye sokarak içindeki elemanlara ulaşıyoruz. Ek olarak find_all methodunu limitleyebilirsiniz. Yani herşeyi getirme sadece 3 tanesini getir diyebilirsiniz, bunun için de **limit=x** gibi bir parametre geçmeniz gerek.

```
source.find_all("a", attrs={"class": "item_1523a"}, limit=2)
```

Peki ya bu linklerin adını değil de yönlendiği adresi çekmek isteseydik? O zaman da **get("href")** değerini kullanmalıyız.

```
>>> for link in solmenu:
...     print(link.get("href"))
...
/
/topics/tech
/topics/games
/topics/books
/topics/artificial-intelligence
/topics/developer-tools
/topics/home
/topics/productivity
/topics/touch-bar-apps
/topics/wearables
/topics
/yours
```

Gördüğünüz gibi döngü içinde her bir link için **get("href")** ile bağlantı adreslerini çektiğimiz. Sadece buraya kadar öğrendiklerimiz ile mesela bir uygulama yazabilirmiz, bu uygulama da bir site içerisindeki tüm linkleri ekrana yazdırır, web sitenizden hangi sayfalarla çıkış yapılabiliyor onları inclemek isteyebilirsiniz.

Peki herhangi bir seçici belirtmeden direkt olarak metin aramak istediğimizde?

```
>>> source.find_all(string=["Sign Up", "Log In"])
['Log In', 'Sign Up']
>>> source.find_all(string=["Sign Up", "Log Inssss"])
['Sign Up']
```

ilk satırda Sign Up ve Log In metinlerini aradım ikisi de olduğu için aynı şekilde geri liste olarak döndü, ikinci örnekte ise Log Inssss aradım, böyle bir metin olmadığı için sadece olan metin Sign Up geri döndü. Bunu bir seçici ile birlikte de kullanabiliriz.

```
>>> source.find_all("span", string="Subscribe")
[<span class="font_9d927 xSmall_1a46e semiBold_e201b buttonContainer_b6eb3 uppercase">
```

Burda ise span etiketi içinde string değeri Subscribe olanları getir dedik.

next_sibling

İngilizcede sibling kardeş anlamına geliyor, burda da sayfa içinde seçtiğiniz html etiketi ile aynı seviyede diğer html etiketini getirir.. Basit bir örnek verelim.

```
<ul>
<b>Test1</b>
<c>test2</c>
</ul>
```

Yukarıdaki örnek de Test1 ile Test2 kardeş çünkü aynı seviyedeler. Şimdi biz kendi örneğimizde bir sonraki kardeş bulalım.

```
>>> source.find("ul", attrs={"class": "list_0372b"}).li.text
'Home'
>>> source.find("ul", attrs={"class": "list_0372b"}).li.next_sibling.text
'Tech'
```

Ben de yine soldaki menüde bulunan linklere ulaşmak istedim, bu linkler bir ul etiketi içindeydi, onun class değerini parametre olarak geçtim sonra da bu ul etiketinin altındaki ilk li değerinin text sonucu aldım. Bu da Home değeriydi. Sonra da next_sibling methodu ile de bu Home değerinden sonraki linki istedim o da geldi Tech olarak.

previous_sibling

Yukarıda anlatılan olayın tam tersi, yani seçilen etiketin kardeşi olan ama bu etiketten bir önce geleni getirir.

next_sibling ve previous_sibling

Bunlar da find ile find_all arasındaki ilişki gibi, önce veya sonra 1 kardeş etiketi değil tüm kardeş etiketi getir anlamında kullanılıyor. Seçtiğiniz etiket ile kardeş altında veya üzerinde olan alanları getirmek istedığınızda kullanabilirsiniz.

select

Css selector kullanarak da seçim yapabilirsiniz, mesela sayfa kaynağındaki sol taraftaki menülerden 3. sünün adını getirelim.

```
>>> get_links = source.find("ul", attrs={"class": "list_0372b"}).select("li:nth-of-type(3)")  
>>> for i in get_links:  
...     print(i.text)  
...  
Games
```

Gördüğü gibi Games geldi. Bir örnek daha yapalım

```
>>> source.select("html > body > main > div > div > div > div > div > a:nth-of-type(1)")[0].text  
<span>The best new products, every day</span>
```

Şimdi de sitenin sloganını çektiğ. En tepeden html etiketinden başlayarak alt etiketlere indik.

```
>>> source.select("p.text_44214")  
<p class="text_44214">Product Hunt surfaces the best new products, every day. It's a  
</p>
```

Kaynak içindeki p etiketlerinden text_44214 class değerine sahip olanı çektiğ.

```
>>> source.select('a[href="/ship"]')  
<a href="/ship">Ship <span style="font-style: normal; font-weight: normal">⚠</span></a>
```

Linkler içerisinde href değeri /ship olanı çektiğ. Bazen de tam olarak bağlantı adresini bilemezsiniz, mesela bağlantı adresi belirli bir şablon ile başlayanları çekmek istersiniz. Örnek verelim;

```
>>> linkler = source.select('a[href^="/topics/t"]')  
>>> for link in linkler:  
...     print(link.text)  
...     print(link.get("href"))  
...  
Tech  
/topics/tech  
Touch Bar Apps  
/topics/touch-bar-apps
```

Burda ise bağlantı adresleri içinde href değeri /topics/t ile başlayanları çektiğ. Bunlar da tech ve touch-bar-apps adresleriymiş.

Buraya kadar HTML ağaç yapısı içerisinde istediğimiz verileri aldık, şimdi de bu verileri modifiye etmeyi görelim.

Html Kodlarını Modifiye Etme

Bir kaynaktaki html kodun kopyasını alıktan sonra bu kopyası üzerinde yapabileceğimiz değişikliklere bakalım.

Bir etiketin id veya class değerini değiştirme

Kaynak içindeki bir li etiketine class veya id değeri atayabilirsiniz. Hemen bir örnek yapalım.

```
>>> source.find("ul", attrs={"class": "list_0372b"}).next  
<li><a class="item_1523a activeItem_c89bf item_1523a" href="/"><div class="greyIcon"...
```

Burda producthunt sitesindeki soldaki menülerden ilki olan Home değerini çektiğ. Dikkat ettiyseñiz gelen değer li etiketi ile başlıyor. Şimdi de bu li değerine class atayalım.

```
>>> source.find("ul", attrs={"class": "list_0372b"}).next["class"] = "changed"  
>>> source.find("ul", attrs={"class": "list_0372b"}).next  
<li class="changed"><a class="item_1523a activeItem_c89bf item_1523a" href="/"><div...
```

Şimdide bu li etiketine changed adında bir class atadık. Tekrar bu li etiketini sorguladığımızda da görüldü. Hadi bir de id ekleyelim bu li etiketine.

```
>>> source.find("ul", attrs={"class": "list_0372b"}).next["id"] = "newId"  
>>> source.find("ul", attrs={"class": "list_0372b"}).next  
<li class="changed" id="newId"><a class="item_1523a activeItem_c89bf item_1523a" href="/"><div...
```

Gördüğünüz gibi newId adında bir id ekledim ve li class="changed" id="newId" şeklinde geldi.

String değerini değiştirmek

Şimdide producthunt sitesindeki soldaki menülerden ilki olan Home değerini Anasaya olarak değiştirelim.

```
>>> source.find("span").attrs={"class": "itemText_063f9"}
```

```
<span class="font_9d927 black_476ed small_231df normal_d2e66 itemText_063f9">Home</span>
>>> source.find("span", attrs={"class": "itemText_063f9"}).string
'Home'
>>> source.find("span", attrs={"class": "itemText_063f9"}).string = "Anasayfa"
>>> source.find("span", attrs={"class": "itemText_063f9"})
<span class="font_9d927 black_476ed small_231df normal_d2e66 itemText_063f9">Anasayfa
```

İlk önce span etiketlerinden belirli bir class değerini çekerek Home bağlantısına ulaştık. Ardından string methodu ile ekrana diret olak html halı olmadan metni bastırıldık ve Home değerine ulaştık. sonra da bu Home değerini Anasayfa olacak şekilde değiştirdik. Sonra tekrar sorguladığımızda Home değerinin artık olmadığını Anasayfa değerinin olduğunu görebildik.

Şimdi de bu Anasayfa yazan yeri.append() methodu ile değiştirelim.

```
>>> source.find("span", attrs={"class": "itemText_063f9"})
<span class="font_9d927 black_476ed small_231df normal_d2e66 itemText_063f9">Anasayfa
>>> source.find("span", attrs={"class": "itemText_063f9"}).append("Adresi")
>>> source.find("span", attrs={"class": "itemText_063f9"})
<span class="font_9d927 black_476ed small_231df normal_d2e66 itemText_063f9">AnasayfaAdresi
```

Anasayfa yazdığını kontrol ettim sonra ise.append("Adresi") ile Anasayfa metninin AnasayfaAdresi olmasını sağladım. Yani append ile bu alandaki metnin yanına ekleme yaptım.

Yeni bir etiket ekleme ve silme

Seçtiğimiz bir html etiketi içerisindeki istersek yeni bir etiket daha ekleyebiliriz.

```
>>> source.find("span", attrs={"class": "itemText_063f9"})
<span class="font_9d927 black_476ed small_231df normal_d2e66 itemText_063f9">Anasayfa
>>>
>>> tag = source.new_tag("b")
>>> tag.string = "Bold Metin"
>>>
>>> source.find("span", attrs={"class": "itemText_063f9"}).string.insert_before(tag)
>>> source.find("span", attrs={"class": "itemText_063f9"}).b
<b>Bold Metin</b>
```

İlk önce AnasayfaAdresi adresini seçtik, sonra tag adında bir değişken içerişine direkt olarak source içine new.tag methodu ile bir b etiketi ekledik. Sonra da tag.string ile bu b etiketinin içerişine ne yazması gerektiğini söylediğim. Şimdi de AnasayfaAdresi metnininden önce insert_before(tag) methodu ile tag değişkenini enjekte ettik. Sonra da b tagı var mı diye kontrol ettik başarıyla geldiğini görebildik.

Burda insert_before() methodunu kullandık, böylece AnasayfaAdresi metninin önüne ekledik, eğer sonrasında eklemek isteseydik insert_after() methodunu kullanmamalıydık.

Bir etiketin içindeki değerleri silmek istersek:

```
>>> source.find("span", attrs={"class": "itemText_063f9"}).b
<b>Bold Metin</b>
>>> source.find("span", attrs={"class": "itemText_063f9"}).b.clear()
>>> source.find("span", attrs={"class": "itemText_063f9"}).b
<b></b>
```

İlk önce az önce eklediğim Bold Metin değerini bulduk, sonra clear() methodunu uyguladık, sonra tekrar sorguladığımızda Bold Metin değerinin silindiğini gördüm.

Peki metni değil de etiket ve içindekileri komple silmek isteseydik?

```
>>> source.find("span", attrs={"class": "itemText_063f9"}).b.decompose()
>>> source.find("span", attrs={"class": "itemText_063f9"}).b
```

O zaman da decompose() methodunu uygulamamız gereklidir, ben uyguladım sonra tekrar b etiketini sorguladım herhangi bir şey dönmedi. Yani sildi.

Çıktı Formatları

BeautifulSoup html kodlarının çıktılarını alırken kullanılmak üzere güzel methodlar barındırır. Hemen inceleyelim;

```
>>> print(source.find("span", attrs={"class": "itemText_063f9"}))
<span class="font_9d927 black_476ed small_231df normal_d2e66 itemText_063f9">Anasayfa
>>> # Şimdi prettyfify methodunu kullanalım.
>>> print(source.find("span", attrs={"class": "itemText_063f9"}).prettyify())
<span class="font_9d927 black_476ed small_231df normal_d2e66 itemText_063f9">
  AnasayfaAdresi
</span>
>>>
```

AnasayfaAdresi olarak değiştirdiğim alanı çektiğim html olarak span etiketi geldi. Şimdi bu html dosyasını daha güzel bir şekilde ekrana bastırmak istedigimizde prettyfify methodunu kullanabiliyoruz. Kullanıldığında html kodlarının nasıl daha okunabilir hale geldiğine dikkat edin. Kullanmadığında yanyana yazıyordu, şimdi daha okunaklı.

Encode Etmek

Buraya kadar hep producthunt sitesi üzerinden işlem yapmıştım. Şimdi bu örnek için elle bir html yazıp onun üzerinden anlatım yapacağım.

Eğer işlem yaparken encode değerini değiştirmek isterseniz bunun için bir method mevcut.

```
>>> kaynak = "<div><p><b>Atatürk</b> Ülkeye Çağ Atlattı.</p></div>"
```

```

>>> source = BeautifulSoup(kaynak,"lxml")
>>> print(source.prettify())
<html>
<body>
<p>
<b>
    Atatürk
</b>
Ülkeye Çağ Atlattı.
</p>
</body>
</html>
>>> source.encode("latin-1")
b'<html><body><div><p><b>Atat&uuml;r</b> \xdclkeye \xc7a&#287; Atlatt&#305;. </p></div>
>>> source.encode("utf-8")
b'<html><body><div><p><b>Atat\xc3\xbcrk</b> \xc3\x99lkeye \xc3\x87a\xc4\x9f Atlatt\xc3\x9f.

```

ilk önce **kaynak** adında bir değişken içeresine html kod yazdım. Sonra **prettyf** methodu ile daha okunaklı hale getirdim. Sonra da **.encode("latin-1")** ve **encode("utf-8")** methodları ile karakterlerin encode değerlerini değiştirebildim.



← ÖNCESİ YAZI

SONRAKİ YAZI →

sinanerdinc.com Yorum İkisi

Yorum yazmak, bu konu hakkında özgürce fikirlerinizi paylaşmanıza olanak tanır.



9 Yorum sinanerdinc.com Disqus Gizlilik İkisi

Oturum Açıñ -

Oner 1

Tweet Gönder

Paylas

En İyile Göre Sırala -



Royal Story - 2 yıl önce

import requests
from bs4 import BeautifulSoup as bs

```

url ="http://www.turkanime.tv/index"
headers_param = {"User-Agent": "....."}
r = requests.get(url,headers=headers_param )
soup = bs(r.content,"lxml")

animeliste = soup.find_all("div",attrs={"class":"btn-group btn-group-sm"})

for anime in animeliste:
    anime_link = (anime.a.get("href"))
    print(anime_link)

```

Yazdığım kodlar böyle, url'de bulunan linkler "<http://www.turkanime.tv/imajlarlo>" bu şekilde yani yanında http: yok ve href="javascript:void(0); diye bir şey çırkyor linkleri çekmek istedigimde. Ne yapmalyim ?

1 ^ | v - Yanıtla - Paylaş



Merve Alpay - 6 ay önce

Harika olmuş, elinize sağlık.

^ | v - Yanıtla - Paylaş



safak saryildiz - 2 yıl önce

merhaba;
yazdırduğum sonucu xml aktarma konusunda destek olabilir misiniz.

^ | v - Yanıtla - Paylaş



facebook.100009005657362 - 2 yıl önce

Merhaba,
Ben imdb top250 listesinden veri çekiyorum ama sitesinde film isimleri turkish görünürken benim çektiğim verilerde ingilizce görünümekte sebebi nedir acaba? İyi çalışmalar

^ | v - Yanıtla - Paylaş



Sinan Erdinç Admin → facebook-100009005657362 - 2 yıl önce

Yazdığınız kodları paylaşırısanız hep beraber kontrol edelim, tahimin şu olur. Sayfa kaynağında ingilizce olarak geliryor film isimleri, ancak javascript ile dili Türkçe yapılmıştır. Siz http isteği attığınızda javascript çalışıramayacağınız için size gelen ingilizce olmasını rağmen, sitede gördüğünüz dili Türkçe olabilir.

^ | v - Yanıtla - Paylaş



King Master - 3 yıl önce

ben gerek videolarınızı gerek blogdaki yazıları okudum ama instagramdan veri çekemiyorum.
acaba bu konuda yardımcı olabilir misiniz? cevabınız bekliyorum

gmail adresimi:trkzmn1@gmail.com

^ | v - Yanıtla - Paylaş



Sinan Erdinç Admin → King Master - 2 yıl önce

Yazdığınız kodları paylaşırısanız, nerede takıldığınızı iletirseniz yardımcı olmaya çalışalım tüm ziyaretçilerimiz ile birlikte.

^ | v - Yanıtla - Paylaş



King Master → Sinan Erdinç - 2 yıl önce

Sorunu webdriver headless ile çözüdm. Ama request ve beautifulsoup ilede öğrenmek isterim. Hızlı ve daha küçük boyutlu olur

^ | v - Yanıtla - Paylaş



King Master → Sinan Erdinç - 2 yıl önce

İlgili sorunu webdriver ile çözüdm. Ama request + beautifulsoup ile de öğrenmek isterim.
Daha hızlı ve düşük boyutlu olur.

^ | v - Yanıtla - Paylaş

Abone Olun

Sitenize Disqus ekleyin

Verilerin Sabitlenmesi

DISQUS



Sinan Erdinç • 2020 • sinanerdinc.com
[sitemap](#)