<https://www.google.com/search?q=bike> ---> search=bike

<https://www.google.com/search?q=bike+bos> ---> search= bike boss

<https://www.google.com/search?q=bike+%2B+boss> ---> search= bike + boss

<https://www.google.com/search?q=bike+%26+boss> ---> search= bike & boss

import requests

from bs4 import BeautifulSoup

url = 'https://en.wikipedia.org/wiki/List\_of\_Game\_of\_Thrones\_episodes'

result = requests.get(url)

contents = BeautifulSoup(result.text , 'html.parser')

contents.find('h2') ==== find first tag h2

contents.findAll('h2')

======= find all tag h2

contents.find\_all('h2')

متدهای find و findAll در beautifulsoup پایتون برای پیدا کردن المان های خاص از میان انبوهی از المان ها به کار میروند

برگرداندن تمام تگ های h1 و h2 (دقت که حتما باید داخل یک لیست قرار بگیرد )

for content in contents.findAll(['h1','h2']):

print(content)

به دلیل اینکه کلمه class در پایتون یک کلمه رزرو شده میباشد برای پیدا کردن بر اساس class باید اینگونه کد بنویسیم :

contents.find('' , class\_='vevent')

با استفاده از re و select شما میتونید المان های مورد نظرتون رو به کمک regex پیدا کنید

result1 = contents.find(re.compile('^d'))

result2 = contents.select('a')

result3 = contents.select('a' , limit=5)

result4 = contents.select('#p-logo')

result5 = contents.select('div#p-logo') # ======>tag div with id:p-logo

result6 = contents.select('tr.vevent') # =======>tag tr with class:vevent

result7 = contents.select('a[href^="/wiki/"]')

result8 = contents.select('ul > li')

در result8 تگ های liای که فرزند مستقیم ul باشد را برمیگرداند

همچنین select\_one هم داریم

تفاوت ظاهری متد post و get این است که متد get از طریق url اطلاعات را ارسال میکند اما متد post یک request جدید میسازد و ازطریق آن اطلاعاتی که داخل params قرار گرفته است را ارسال میکند . از طریق متد get امکان ارسال عکس و فیلم وجود ندارد و url محدودیت سایز دارد اما متد پست اینگونه محدود نمیباشد

result = requests.post(url , data=formsData)

result = requests.get(url , params=formsData)

headerها اطلاعات اضافی هستند که مرورگر به سرور ارسال میکنه تا سرور بفهمه که شما کی هستید و به چه اطلاعاتی نیاز دارید

کوکی ها اطلاعاتی هستند که سرورها داخل مرورگر شما ذخیره میکنن تا دفعه بعدی که وارد اون وبسایت شدید سرور شما رو بشناسه و دیگه مجبور نشید دوباره اطلاعات تکراری رو به سرور ارسال کنید

Path=/ به این معنی هست که در تمام مسیر ها این کوکی قابل استفاده هست

myCookies ={

    'PHPSESSID':request\_post.cookies['PHPSESSID']

}

request\_get = requests.get(url + 'secret.php' , cookies=myCookies)

بعضی وقت ها ممکن است به صورت مستقیم شما به صفحه secret ریدایرکت شوید در این صورت نمیتوانید به کوکی ها دسترسی پیدا کنید برای حل این مشکل باید کار زیر را انجام دهید :

request\_post = requests.post(url, allow\_redirects=False)

sessionها اطلاعاتی هستند که سرورها به صورت موقت در مرورگر شما ذخیره میکنن تا فعالیت هاتون رو تحت نظر داشته باشن

خوبی sessionها در پایتون این است که کاری با cookieها و redirectها ندارد و خوده session آنها را هندل میکند تنها کاری که لازم است انجام شود ساخت یک object از Session میباشد به این صورت انجام میشود

MySession = requests.Session()

در sessionها هم به راحتی میتوانیم به get و post دسترسی داشته باشیم

برای update کردن headerها در session ها کافیست از روش زیر استفاده شود :

MySession.headers.update({'User-Agent':'sajjad'})

Selenium ابزاریست برای اتوماتیک کردن مرورگرها. هدف اصلی استفاده از این ابزار تست برنامه های وب است.

implicit wait به این معنی است که مرورگر باید تا زمان خاصی صبر کرده تا عملی انجام شود، بعد از آن مرورگر میتواند به کار خود ادامه دهد

به دلیل اینکه مرورگر بتواند اطلاعات لازم را لود کند مقداری زمان لازم است حال ما باید برنامه پایتون را مجبور کنید مقداری زمان صبر کند برای اینکار میتوانیم از implicitly\_wait استفاده کنیم (عددی که داخل پرانتز نوشته میشود حداکثر زمانی است که باید صبر کند یعنی اگر لود صفحه کمتر از این زمان بود خود برنامه دیگر صبر نمیکند )

driver.implicitly\_wait(10)

explicit wait هم مانند implicit wait میتونه مرورگر رو مجبور به صبر کردن کنه اما explicit wait قدرت و استفاده بیشتری داره

……..