review_in_json_format

March 18, 2020

[3]:
```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import urllib.request
import urllib.parse
import urllib.error
from bs4 import BeautifulSoup
import ssl
import json


# For ignoring SSL certificate errors
ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE

url=input("Enter Amazon Product Url- ")
html = urllib.request.urlopen(url, context=ctx).read()
soup = BeautifulSoup(html, 'html.parser')
html = soup.prettify('utf-8')
product_json = {}
# This block of code will help extract the Brand of the item
for divs in soup.findAll('div', attrs={'class': 'a-box-group'}):
    try:
        product_json['brand'] = divs['data-brand']
        break
    except:
        pass
# This block of code will help extract the Prodcut Title of the item
for spans in soup.findAll('span', attrs={'id': 'productTitle'}):
    name_of_product = spans.text.strip()
    product_json['name'] = name_of_product
    break
# This block of code will help extract the price of the item in dollars
for divs in soup.findAll('div'):
    try:
        price = str(divs['data-asin-price'])
        product_json['price'] = '$' + price
```

```
        break
    except:
        pass
# This block of code will help extract the image of the item in dollars

for divs in soup.findAll('div', attrs={'id': 'rwImages_hidden'}):
    for img_tag in divs.findAll('img', attrs={'style': 'display:none;'
                                    }):
        product_json['img-url'] = img_tag['src']
        break
# This block of code will help extract the average star rating of the product
for i_tags in soup.findAll('i',
                           attrs={'data-hook': 'average-star-rating'}):
    for spans in i_tags.findAll('span', attrs={'class': 'a-icon-alt'}):
        product_json['star-rating'] = spans.text.strip()
        break
# This block of code will help extract the number of customer reviews of the␣
 ↪product
for spans in soup.findAll('span', attrs={'id': 'acrCustomerReviewText'
                          }):
    if spans.text:
        review_count = spans.text.strip()
        product_json['customer-reviews-count'] = review_count
        break
# This block of code will help extract top specifications and details of the␣
 ↪product
product_json['details'] = []
for ul_tags in soup.findAll('ul',
                            attrs={'class': 'a-unordered-list a-vertical␣
 ↪a-spacing-none'
                            }):
    for li_tags in ul_tags.findAll('li'):
        for spans in li_tags.findAll('span',
                attrs={'class': 'a-list-item'}, text=True,
                recursive=False):
            product_json['details'].append(spans.text.strip())

# This block of code will help extract the short reviews of the product

product_json['short-reviews'] = []
for a_tags in soup.findAll('a',
                           attrs={'class': 'a-size-base a-link-normal␣
 ↪review-title a-color-base a-text-bold'
                           }):
    short_review = a_tags.text.strip()
    product_json['short-reviews'].append(short_review)
# This block of code will help extract the long reviews of the product
```

```
product_json['long-reviews'] = []
for divs in soup.findAll('div', attrs={'data-hook': 'review-collapsed'
                         }):
    long_review = divs.text.strip()
    product_json['long-reviews'].append(long_review)
# Saving the scraped html file
with open('output_file.html', 'wb') as file:
    file.write(html)
# Saving the scraped data in json format
with open('product.json', 'w') as outfile:
    json.dump(product_json, outfile, indent=4)
print ('----------Extraction of data is complete. Check json file.----------')
```

```
Enter Amazon Product Url- https://www.amazon.in/gp/product/B07HGJKDQL?pf_rd_r=DN
C0ENPN86QSKS65S6V0&pf_rd_p=649eac15-05ce-45c0-86ac-3e413b8ba3d4
----------Extraction of data is complete. Check json file.----------
```