eBay Data Processing and Extraction for Sellers in Marketplace

Scope:

eBay Seller Price Estimator –

Our project idea came with the thought of private sellers. For private sellers there are many dynamic market factors that may change the best-selling price of a given item, especially when in competition with large corporations and wholesalers. What would be the most reliable reference tools to determine what the price for a given item should be? What can an individual seller expect in terms of best price if they want to sell now, or if they're willing to wait to get that best price? A lot of these questions can be answered by information that’s already available, the recent market history of a given item. Our project aims to make this market history information more readily available to a seller for comparisons across not only eBay, but across any other marketing platform.

Why:

Why did we decide to choose eBay?

We chose eBay because of the "raw" information that the company provides. In comparison to its largest, and superior competitor, Amazon; eBay is much more unstructured (pre-sorted per say) when it comes to responding to user searches. We suspect that, among a variety of reasons, a strength of Amazon item search engine may be how it sorts and responds to user searches. However, we believe a user may benefit from using eBay, rather than Amazon, when selling items. With these items we have extracted (sellers) from eBay, we want to compare prices along to eBay's adversary, Amazon.

-When a user searches for an item under Amazon, the user receives the same item, from varying sellers, for varying prices; and Amazon presents what it calculated to be the best price for that item (pre-sorted or the user). All other listings for a particular item appear as sub listings under that top item!
In Amazon you can easily find out which is the cheapest, as it is likely the first, second, or third item that was displayed.

- When a user searches for an item under eBay, the user receives various items/listings from different sellers with no form of "pre-sorted" displaying method (other than sponsored top listings, which are rarely the best price). In eBay, a better price for a buyer or user may be found with a little bit of "digging", and that is what we are attempting to do with our project. We want to allow the user to "filter" by a few factors that presently are not available on Amazon or eBay, and to reveal information about a particular item market that may be important to a seller.

We believe users who are very active traders, may be overpaying for items on Amazon (although hard to compete with Amazon's distribution network, value yet undetermined), and losing money if selling on Amazon.

Variables of Data:

With many items to choose from on eBay, we sat down and thought a lot about what items sellers are having the most luck with. And which items would appeal to a pro-seller. we chose these items of interest randomly, for ultimate aim, we will not sort this in a file, it will be uploaded online, for the purpose of this course there is no point to select these items.

We chose items such as:

Beats Headphones/JBL Speakers - items that are commonly found in university setting

KitchenAid - household items that are also used very often in any environment

iBlason - cases for any type of phone, laptop

Apple Watch - watch that is popular with students

Xbox - gaming console that many students use frequently

Citizen Watch - other watches that people may have

The items we chose, are not limited to the types of condition they are in,
we also have different conditions that the items may be in.

Conditions range from:

New items - A brand-new, unused, unopened, unworn, undamaged item. Most categories support this condition (as long as condition is an applicable concept).

New other items - A brand-new new, unused item with no signs of wear. Packaging may be missing or opened. The item may be a factory second or have defects.

New with defects - A brand-new, unused, and unworn item. The item may have cosmetic defects, and/or may contain mismarked tags (e.g., incorrect size tags from the manufacturer). Packaging may be missing or opened. The item may be a new factory second or irregular.

Manufacturer refurbished - An item in excellent condition that has been professionally restored to working order by a manufacturer or manufacturer-approved vendor. The item may or may not be in the original packaging.

Seller refurbished - An item that has been restored to working order by the eBay seller or a third party who is not approved by the manufacturer. This means the seller indicates that the item is in full working order and is in excellent condition. The item may or may not be in original packaging.

Used - An item that has been used previously. The item may have some signs of cosmetic wear, but is fully operational and functions as intended. This item may be a floor model or store return that has been used. Most categories support this condition (as long as condition is an applicable concept).

Very Good - An item that is used but still in very good condition. No obvious damage to the cover or jewel case. No missing or damaged pages or liner notes. The instructions (if applicable) are included in the box. May have very minimal identifying marks on the inside cover. Very minimal wear and tear.

Good - An item in used but good condition. May have minor external damage including scuffs, scratches, or cracks but no holes or tears. For books, liner notes, or instructions, the majority of pages have minimal damage or markings and no missing pages.

Acceptable - An item with obvious or significant wear, but still operational. For books, liner notes, or instructions, the item may have some damage to the cover but the integrity is still intact. Instructions and/or box may be missing. For books, possible writing in margins, etc., but no missing pages or anything that would compromise the legibility or understanding of the text.

For parts or not working - An item that does not function as intended and is not fully operational. This includes items that are defective in ways that render them difficult to use, items that require service or repair, or items missing essential components. Supported in categories where parts or non-working items are of interest to people who repair or collect related items.

Besides condition our items also showcase

-what country its coming from.

-UPC and item ID

-Listing Info - which may show how many people are watching the item, if you can buy it now

-payment method accepted

-zip code

-primary category ID

-If returns are accepted

-the selling status

-shipping info

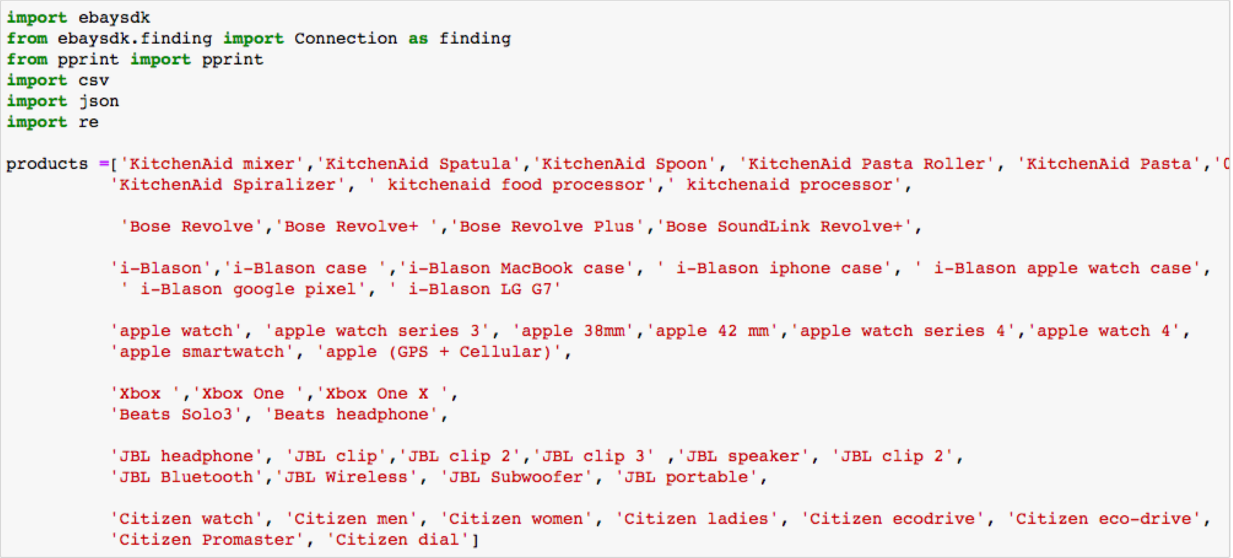
-topRatedListing Item

API:

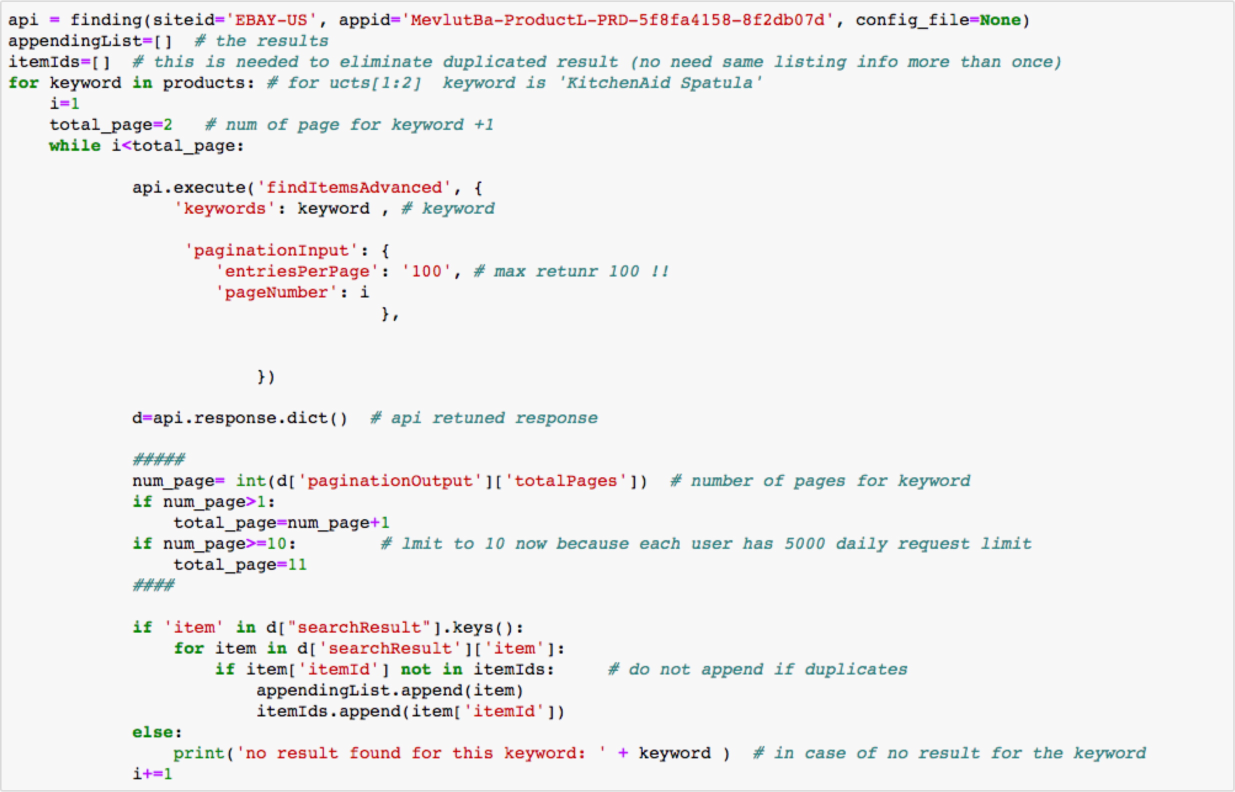
The way we accessed eBay's website and information was through their API.

To get permission to use eBay's API we had to create an account and sign up for their eBay Developer program. The access to eBay's API was fairly simple and we do not need to have special permission for developer API (Finding API, shopping API etc.) unless you do not need the trading API or buyer API. We used Finding API in this project to fulfill this course requirement. It is worth to note that Finding API has 5000 call limit per day for all application. If your application approved (Compatible Application), the limit would be 1.5 million request per day. We prefer to abide 5000 limit and searched some items we chose for the demonstration.

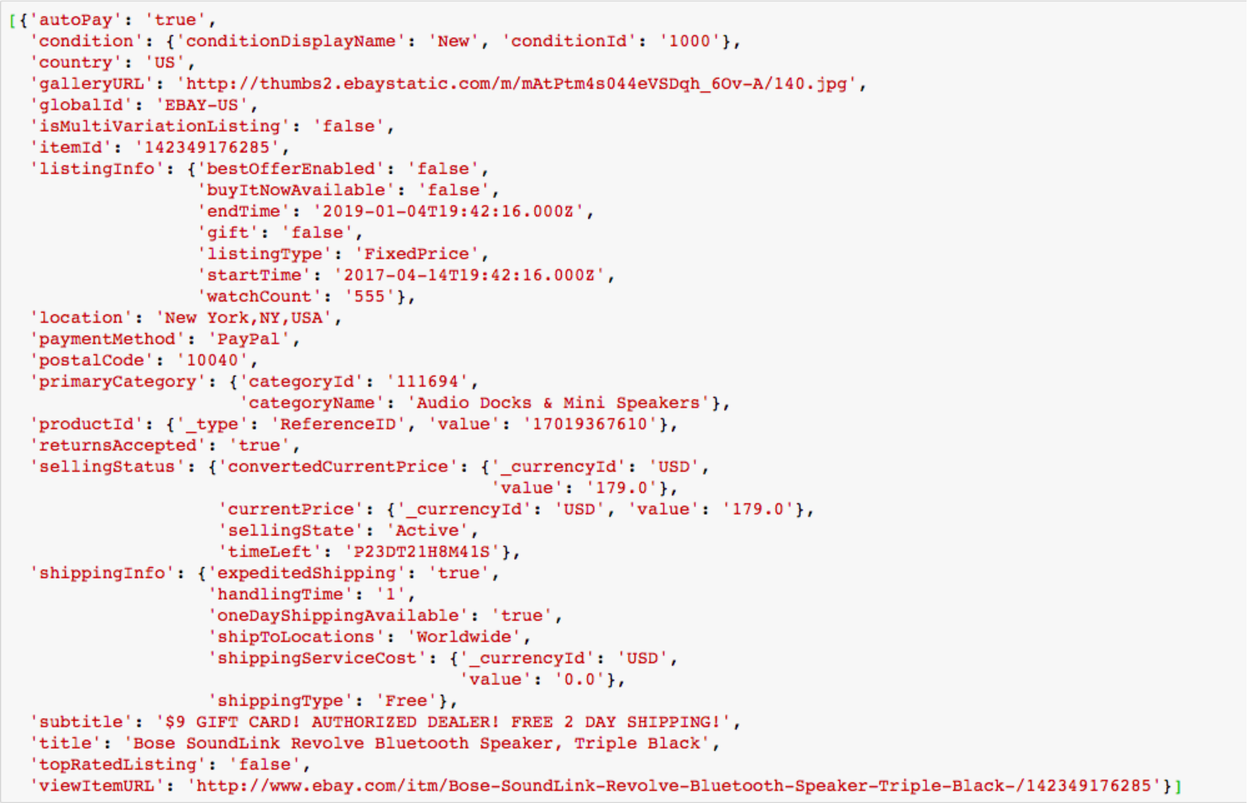
We would also like to mention ebaysdk python toolbox, an open source program, utilize the request results easily. However, the returned results are default results which do not include item UPC number (if available), item quantity and sold item quantity etc. We spent quite a lot amount of time gathering this information by using different API, but eBay does not provide this information in their API.



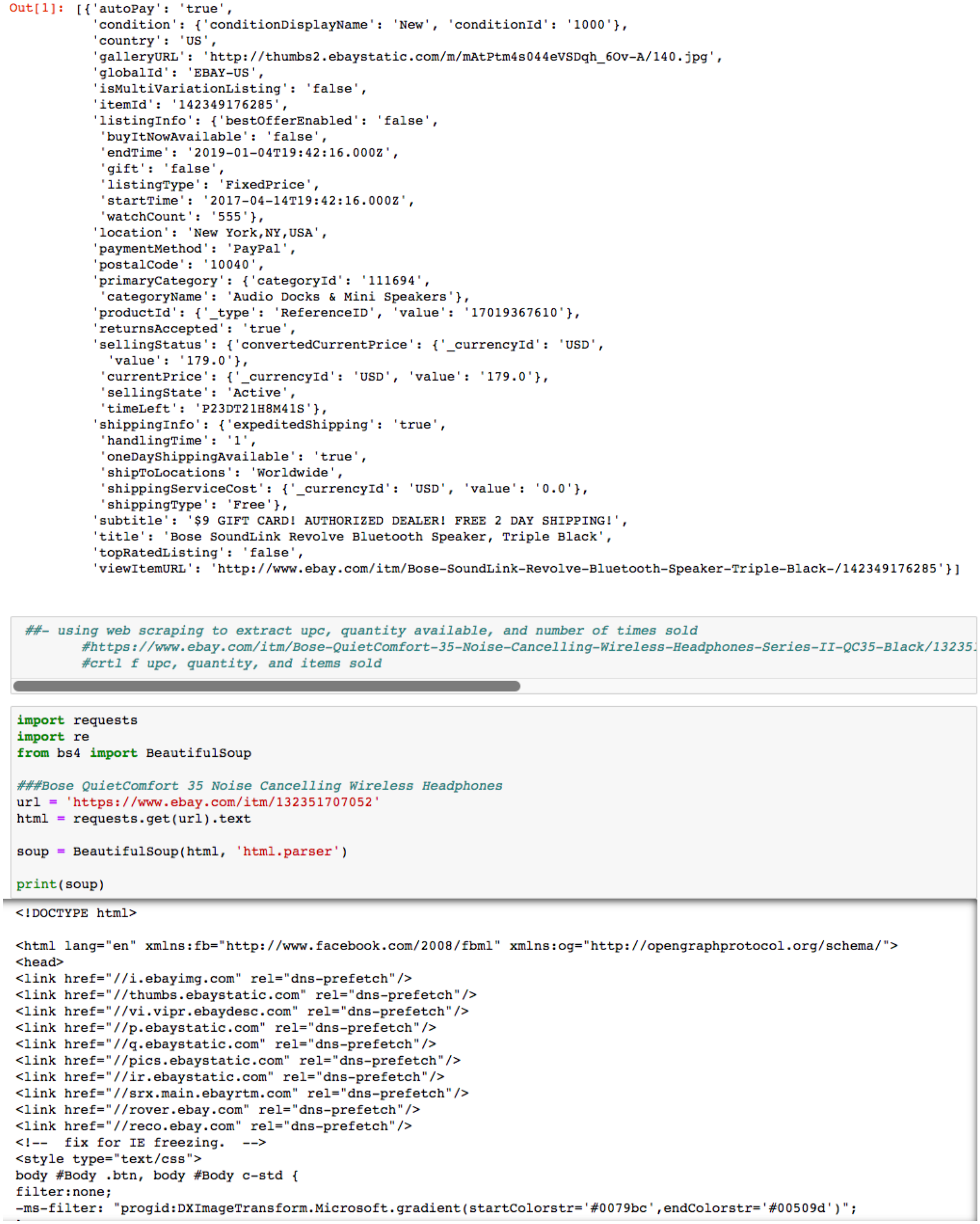
```
import ebaysdk
from ebaysdk.finding import Connection as finding
from pprint import pprint
import csv
import json
import re

products =['KitchenAid mixer','KitchenAid Spatula','KitchenAid Spoon', 'KitchenAid Pasta Roller', 'KitchenAid Pasta','C
          'KitchenAid Spiralizer', ' kitchenaid food processor',' kitchenaid processor',

           'Bose Revolve','Bose Revolve+ ','Bose Revolve Plus','Bose SoundLink Revolve+',

          'i-Blason','i-Blason case ','i-Blason MacBook case', ' i-Blason iphone case', ' i-Blason apple watch case',
           ' i-Blason google pixel', ' i-Blason LG G7'

          'apple watch', 'apple watch series 3', 'apple 38mm','apple 42 mm','apple watch series 4','apple watch 4',
          'apple smartwatch', 'apple (GPS + Cellular)',

          'Xbox ','Xbox One ','Xbox One X ',
          'Beats Solo3', 'Beats headphone',

          'JBL headphone', 'JBL clip','JBL clip 2','JBL clip 3' ,'JBL speaker', 'JBL clip 2',
          'JBL Bluetooth','JBL Wireless', 'JBL Subwoofer', 'JBL portable',

          'Citizen watch', 'Citizen men', 'Citizen women', 'Citizen ladies', 'Citizen ecodrive', 'Citizen eco-drive',
          'Citizen Promaster', 'Citizen dial']
```



```
api = finding(siteid='EBAY-US', appid='MevlutBa-ProductL-PRD-5f8fa4158-8f2db07d', config_file=None)
appendingList=[]  # the results
itemIds=[]  # this is needed to eliminate duplicated result (no need same listing info more than once)
for keyword in products: # for ucts[1:2]  keyword is 'KitchenAid Spatula'
    i=1
    total_page=2   # num of page for keyword +1
    while i<total_page:

            api.execute('findItemsAdvanced', {
                'keywords': keyword , # keyword

                 'paginationInput': {
                    'entriesPerPage': '100', # max retunr 100 !!
                    'pageNumber': i
                                    },


                        })

            d=api.response.dict()  # api retuned response

            #####
            num_page= int(d['paginationOutput']['totalPages'])  # number of pages for keyword
            if num_page>1:
                total_page=num_page+1
            if num_page>=10:        # lmit to 10 now because each user has 5000 daily request limit
                total_page=11
            ####

            if 'item' in d["searchResult"].keys():
                for item in d['searchResult']['item']:
                    if item['itemId'] not in itemIds:     # do not append if duplicates
                        appendingList.append(item)
                        itemIds.append(item['itemId'])
            else:
                print('no result found for this keyword: ' + keyword )  # in case of no result for the keyword
            i+=1
```



```
[{'autoPay': 'true',
  'condition': {'conditionDisplayName': 'New', 'conditionId': '1000'},
  'country': 'US',
  'galleryURL': 'http://thumbs2.ebaystatic.com/m/mAtPtm4s044eVSDqh_6Ov-A/140.jpg',
  'globalId': 'EBAY-US',
  'isMultiVariationListing': 'false',
  'itemId': '142349176285',
  'listingInfo': {'bestOfferEnabled': 'false',
                  'buyItNowAvailable': 'false',
                  'endTime': '2019-01-04T19:42:16.000Z',
                  'gift': 'false',
                  'listingType': 'FixedPrice',
                  'startTime': '2017-04-14T19:42:16.000Z',
                  'watchCount': '555'},
  'location': 'New York,NY,USA',
  'paymentMethod': 'PayPal',
  'postalCode': '10040',
  'primaryCategory': {'categoryId': '111694',
                      'categoryName': 'Audio Docks & Mini Speakers'},
  'productId': {'_type': 'ReferenceID', 'value': '17019367610'},
  'returnsAccepted': 'true',
  'sellingStatus': {'convertedCurrentPrice': {'_currencyId': 'USD',
                                              'value': '179.0'},
                    'currentPrice': {'_currencyId': 'USD', 'value': '179.0'},
                    'sellingState': 'Active',
                    'timeLeft': 'P23DT21H8M41S'},
  'shippingInfo': {'expeditedShipping': 'true',
                   'handlingTime': '1',
                   'oneDayShippingAvailable': 'true',
                   'shipToLocations': 'Worldwide',
                   'shippingServiceCost': {'_currencyId': 'USD',
                                           'value': '0.0'},
                   'shippingType': 'Free'},
  'subtitle': '$9 GIFT CARD! AUTHORIZED DEALER! FREE 2 DAY SHIPPING!',
  'title': 'Bose SoundLink Revolve Bluetooth Speaker, Triple Black',
  'topRatedListing': 'false',
  'viewItemURL': 'http://www.ebay.com/itm/Bose-SoundLink-Revolve-Bluetooth-Speaker-Triple-Black-/142349176285'}]
```

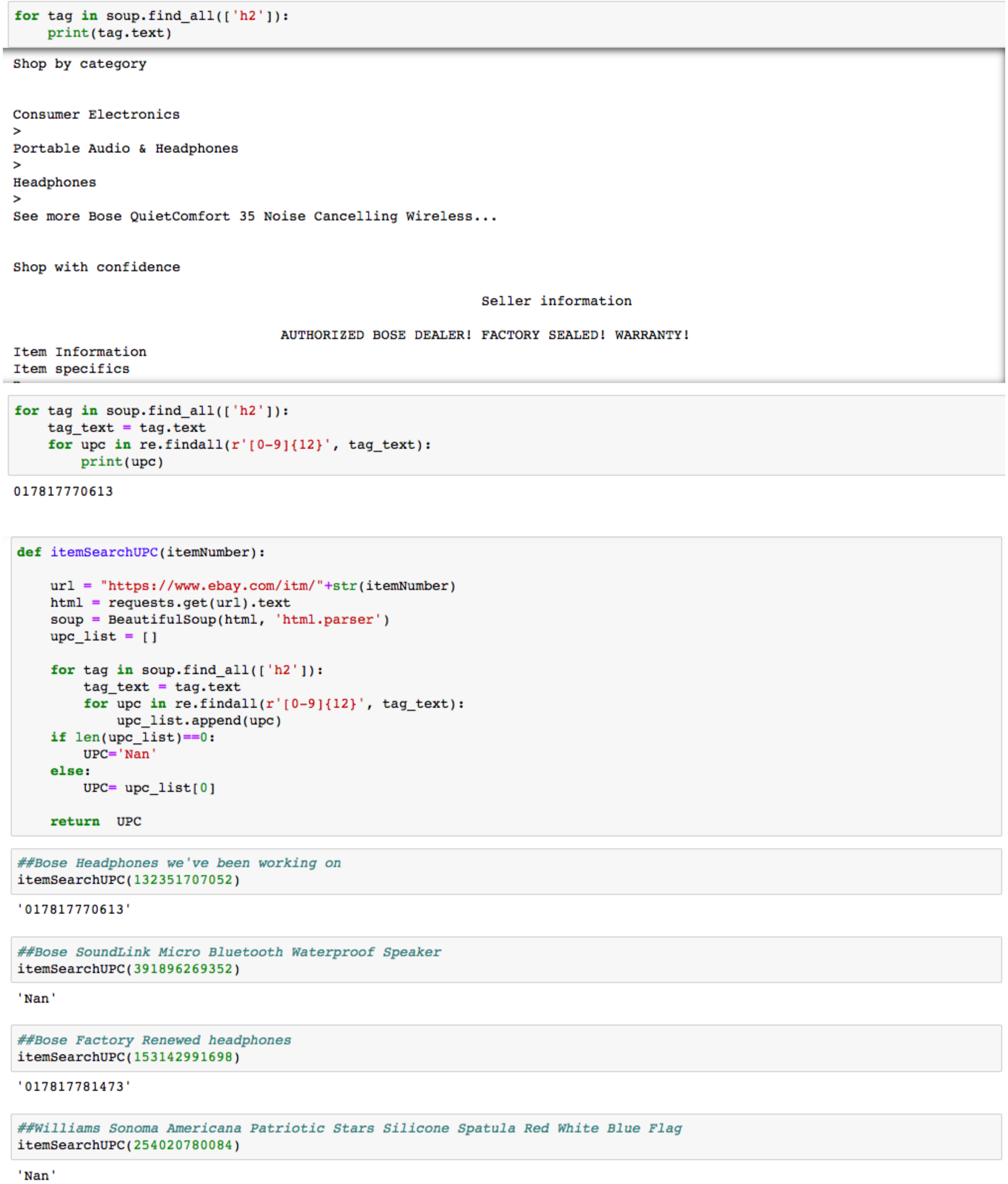


```
Out[1]: [{'autoPay': 'true',
          'condition': {'conditionDisplayName': 'New', 'conditionId': '1000'},
          'country': 'US',
          'galleryURL': 'http://thumbs2.ebaystatic.com/m/mAtPtm4s044eVSDqh_6Ov-A/140.jpg',
          'globalId': 'EBAY-US',
          'isMultiVariationListing': 'false',
          'itemId': '142349176285',
          'listingInfo': {'bestOfferEnabled': 'false',
           'buyItNowAvailable': 'false',
           'endTime': '2019-01-04T19:42:16.000Z',
           'gift': 'false',
           'listingType': 'FixedPrice',
           'startTime': '2017-04-14T19:42:16.000Z',
           'watchCount': '555'},
          'location': 'New York,NY,USA',
          'paymentMethod': 'PayPal',
          'postalCode': '10040',
          'primaryCategory': {'categoryId': '111694',
           'categoryName': 'Audio Docks & Mini Speakers'},
          'productId': {'_type': 'ReferenceID', 'value': '17019367610'},
          'returnsAccepted': 'true',
          'sellingStatus': {'convertedCurrentPrice': {'_currencyId': 'USD',
            'value': '179.0'},
           'currentPrice': {'_currencyId': 'USD', 'value': '179.0'},
           'sellingState': 'Active',
           'timeLeft': 'P23DT21H8M41S'},
          'shippingInfo': {'expeditedShipping': 'true',
           'handlingTime': '1',
           'oneDayShippingAvailable': 'true',
           'shipToLocations': 'Worldwide',
           'shippingServiceCost': {'_currencyId': 'USD', 'value': '0.0'},
           'shippingType': 'Free'},
          'subtitle': '$9 GIFT CARD! AUTHORIZED DEALER! FREE 2 DAY SHIPPING!',
          'title': 'Bose SoundLink Revolve Bluetooth Speaker, Triple Black',
          'topRatedListing': 'false',
          'viewItemURL': 'http://www.ebay.com/itm/Bose-SoundLink-Revolve-Bluetooth-Speaker-Triple-Black-/142349176285'}]
```

```
##- using web scraping to extract upc, quantity available, and number of times sold
        #https://www.ebay.com/itm/Bose-QuietComfort-35-Noise-Cancelling-Wireless-Headphones-Series-II-QC35-Black/13235
        #crtl f upc, quantity, and items sold
```

```
import requests
import re
from bs4 import BeautifulSoup

###Bose QuietComfort 35 Noise Cancelling Wireless Headphones
url = 'https://www.ebay.com/itm/132351707052'
html = requests.get(url).text

soup = BeautifulSoup(html, 'html.parser')

print(soup)
```

```
<!DOCTYPE html>

<html lang="en" xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://opengraphprotocol.org/schema/">
<head>
<link href="//i.ebayimg.com" rel="dns-prefetch"/>
<link href="//thumbs2.ebaystatic.com" rel="dns-prefetch"/>
<link href="//vi.vipr.ebaydesc.com" rel="dns-prefetch"/>
<link href="//p.ebaystatic.com" rel="dns-prefetch"/>
<link href="//q.ebaystatic.com" rel="dns-prefetch"/>
<link href="//pics.ebaystatic.com" rel="dns-prefetch"/>
<link href="//ir.ebaystatic.com" rel="dns-prefetch"/>
<link href="//srx.main.ebayrtm.com" rel="dns-prefetch"/>
<link href="//rover.ebay.com" rel="dns-prefetch"/>
<link href="//reco.ebay.com" rel="dns-prefetch"/>
<!--  fix for IE freezing.  -->
<style type="text/css">
body #Body .btn, body #Body c-std {
filter:none;
-ms-filter: "progid:DXImageTransform.Microsoft.gradient(startColorstr='#0079bc',endColorstr='#00509d')";
```



```python
for tag in soup.find_all(['h2']):
    print(tag.text)
```

```
Shop by category


Consumer Electronics
>
Portable Audio & Headphones
>
Headphones
>
See more Bose QuietComfort 35 Noise Cancelling Wireless...


Shop with confidence

                                                        Seller information

                                AUTHORIZED BOSE DEALER! FACTORY SEALED! WARRANTY!
Item Information
Item specifics
```

```python
for tag in soup.find_all(['h2']):
    tag_text = tag.text
    for upc in re.findall(r'[0-9]{12}', tag_text):
        print(upc)
```

```
017817770613
```

```python
def itemSearchUPC(itemNumber):

    url = "https://www.ebay.com/itm/"+str(itemNumber)
    html = requests.get(url).text
    soup = BeautifulSoup(html, 'html.parser')
    upc_list = []

    for tag in soup.find_all(['h2']):
        tag_text = tag.text
        for upc in re.findall(r'[0-9]{12}', tag_text):
            upc_list.append(upc)
    if len(upc_list)==0:
        UPC='Nan'
    else:
        UPC= upc_list[0]

    return  UPC
```

```python
##Bose Headphones we've been working on
itemSearchUPC(132351707052)
```

```
'017817770613'
```

```python
##Bose SoundLink Micro Bluetooth Waterproof Speaker
itemSearchUPC(391896269352)
```

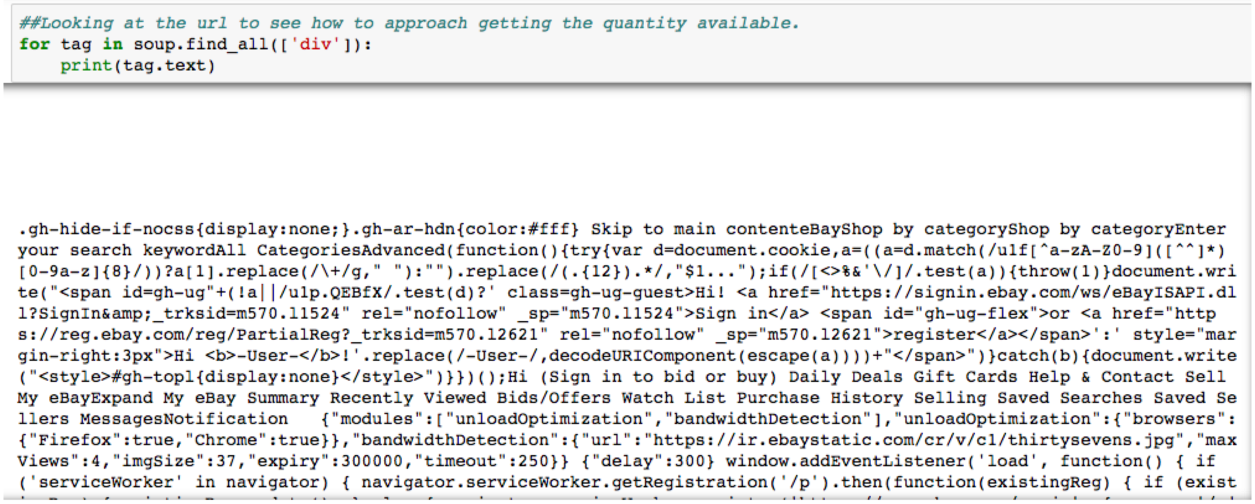
```
'Nan'
```

```python
##Bose Factory Renewed headphones
itemSearchUPC(153142991698)
```

```
'017817781473'
```

```python
##Williams Sonoma Americana Patriotic Stars Silicone Spatula Red White Blue Flag
itemSearchUPC(254020780084)
```

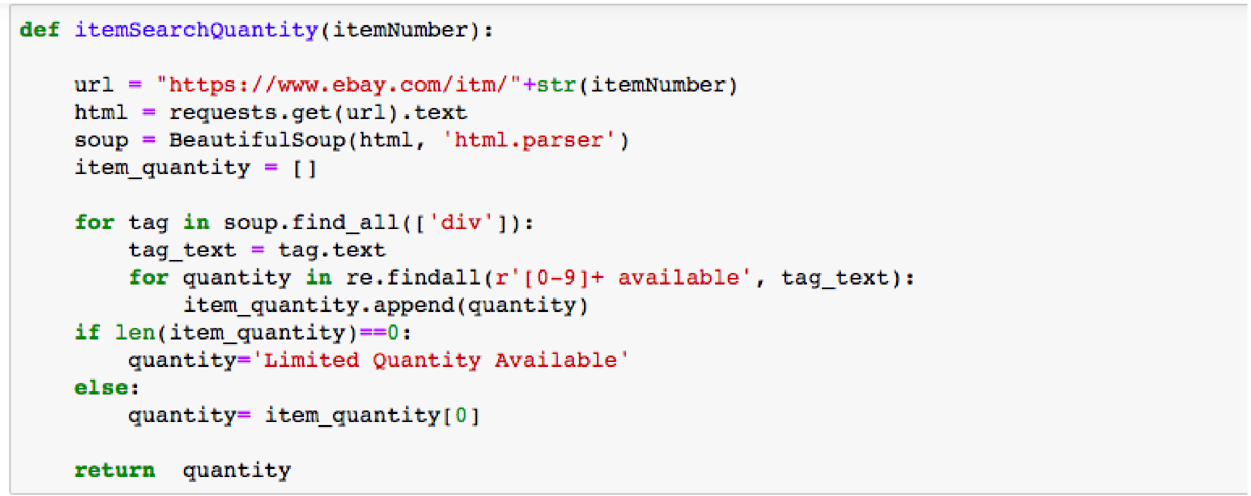
```
'Nan'
```



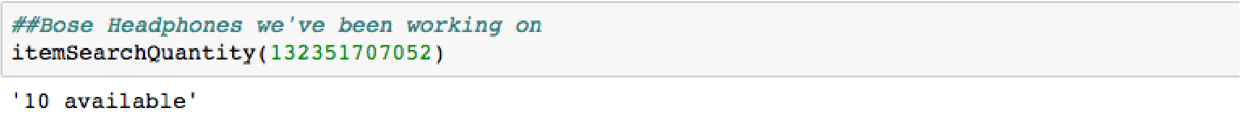
```
##Looking at the url to see how to approach getting the quantity available.
for tag in soup.find_all(['div']):
    print(tag.text)
```

```




.gh-hide-if-nocss{display:none;}.gh-ar-hdn{color:#fff} Skip to main contenteBayShop by categoryShop by categoryEnter
your search keywordAll CategoriesAdvanced(function(){try{var d=document.cookie,a=((a=d.match(/u1f[^a-zA-Z0-9]([^^]*)
[0-9a-z]{8}/))?a[1].replace(/\+/g," "):"").replace(/(.{12}).*/,"$1...");if(/[<>%&'\/]/.test(a)){throw(1)}document.wri
te("<span id=gh-ug"+(!a||/u1p.QEBfX/.test(d)?' class=gh-ug-guest>Hi! <a href="https://signin.ebay.com/ws/eBayISAPI.dl
l?SignIn&amp;_trksid=m570.l1524" rel="nofollow" _sp="m570.l1524">Sign in</a> <span id="gh-ug-flex">or <a href="http
s://reg.ebay.com/reg/PartialReg?_trksid=m570.l2621" rel="nofollow" _sp="m570.l2621">register</a></span>':' style="mar
gin-right:3px">Hi <b>-User-</b>!'.replace(/-User-/,decodeURIComponent(escape(a))))+"</span>")}catch(b){document.write
("<style>#gh-top1{display:none}</style>")}})();Hi (Sign in to bid or buy) Daily Deals Gift Cards Help & Contact Sell
My eBayExpand My eBay Summary Recently Viewed Bids/Offers Watch List Purchase History Selling Saved Searches Saved Se
llers MessagesNotification   {"modules":["unloadOptimization","bandwidthDetection"],"unloadOptimization":{"browsers":
{"Firefox":true,"Chrome":true}},"bandwidthDetection":{"url":"https://ir.ebaystatic.com/cr/v/c1/thirtysevens.jpg","max
Views":4,"imgSize":37,"expiry":300000,"timeout":250}} {"delay":300} window.addEventListener('load', function() { if
('serviceWorker' in navigator) { navigator.serviceWorker.getRegistration('/p').then(function(existingReg) { if (exist
```

The next function created is called – itemSearchQuantity, this searches quantity of items available for purchase just by searching for itemNumber.

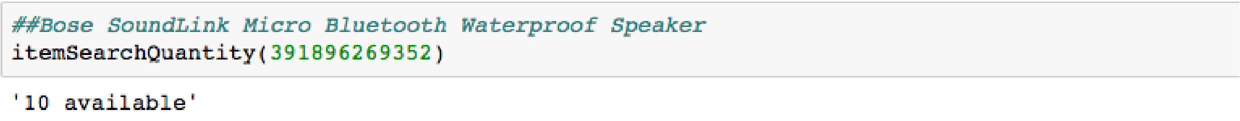


```
def itemSearchQuantity(itemNumber):

    url = "https://www.ebay.com/itm/"+str(itemNumber)
    html = requests.get(url).text
    soup = BeautifulSoup(html, 'html.parser')
    item_quantity = []

    for tag in soup.find_all(['div']):
        tag_text = tag.text
        for quantity in re.findall(r'[0-9]+ available', tag_text):
            item_quantity.append(quantity)
    if len(item_quantity)==0:
        quantity='Limited Quantity Available'
    else:
        quantity= item_quantity[0]

    return  quantity
```



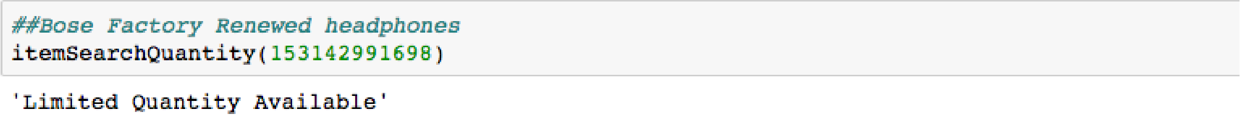
```
##Bose Headphones we've been working on
itemSearchQuantity(132351707052)
```

```
'10 available'
```



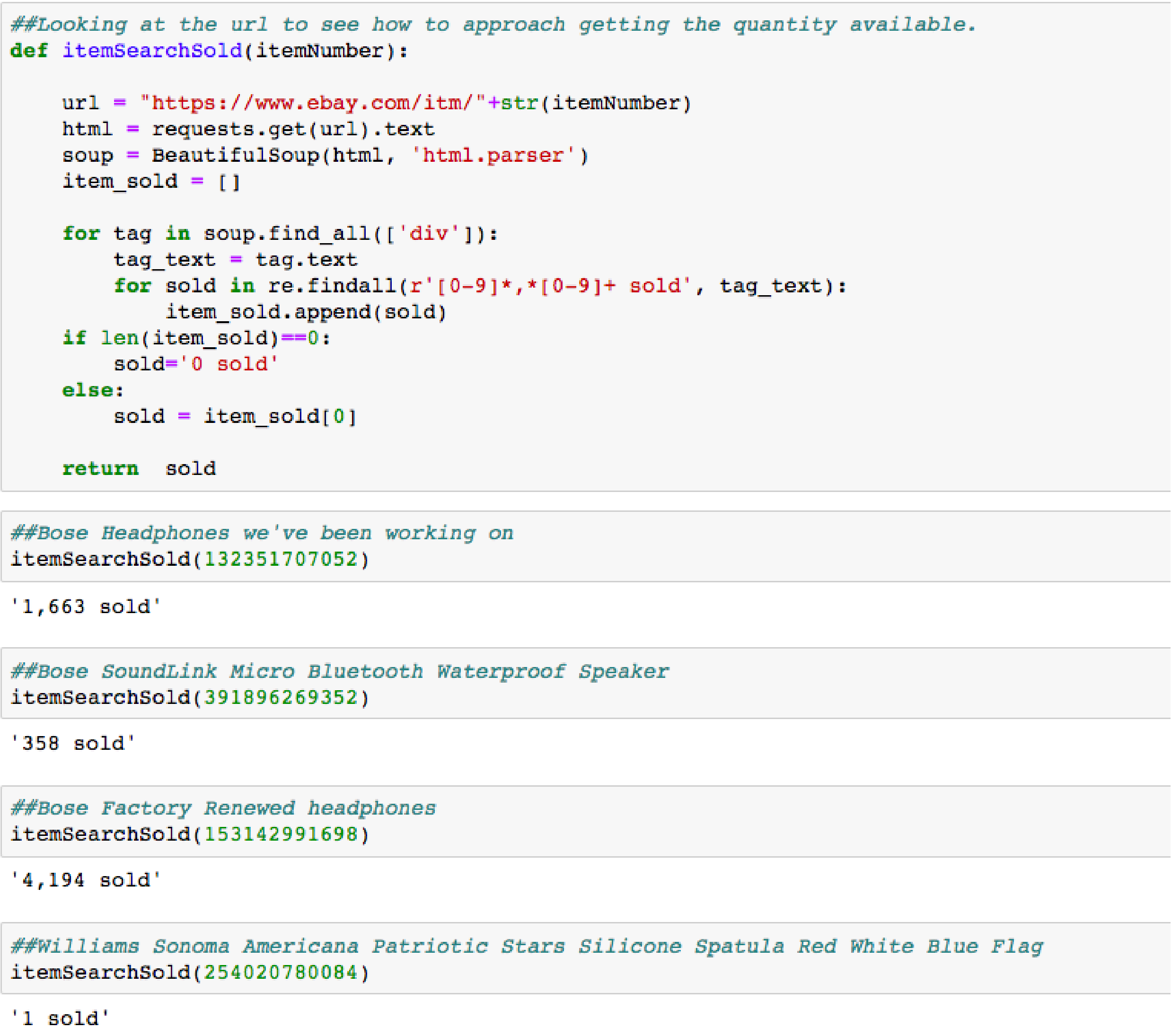
```
##Bose SoundLink Micro Bluetooth Waterproof Speaker
itemSearchQuantity(391896269352)
```

```
'10 available'
```



```
##Bose Factory Renewed headphones
itemSearchQuantity(153142991698)
```

```
'Limited Quantity Available'
```



```
##Looking at the url to see how to approach getting the quantity available.
def itemSearchSold(itemNumber):

    url = "https://www.ebay.com/itm/"+str(itemNumber)
    html = requests.get(url).text
    soup = BeautifulSoup(html, 'html.parser')
    item_sold = []

    for tag in soup.find_all(['div']):
        tag_text = tag.text
        for sold in re.findall(r'[0-9]*,*[0-9]+ sold', tag_text):
            item_sold.append(sold)
    if len(item_sold)==0:
        sold='0 sold'
    else:
        sold = item_sold[0]

    return  sold
```

```
##Bose Headphones we've been working on
itemSearchSold(132351707052)
```

```
'1,663 sold'
```

```
##Bose SoundLink Micro Bluetooth Waterproof Speaker
itemSearchSold(391896269352)
```

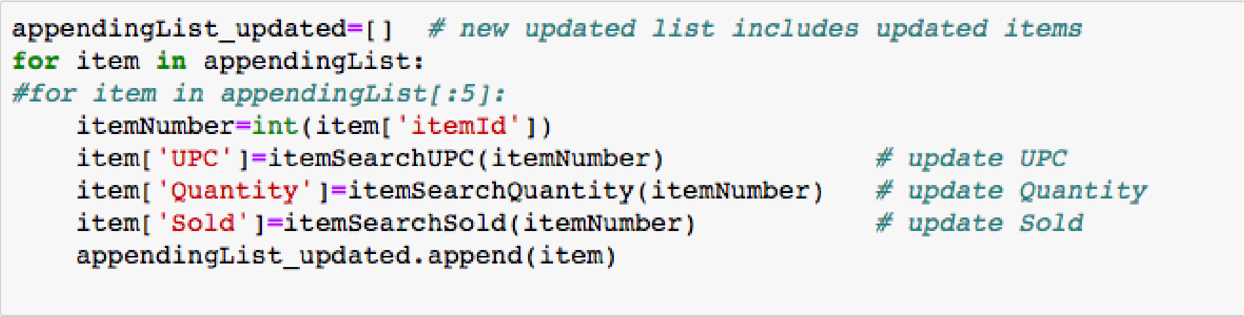
```
'358 sold'
```

```
##Bose Factory Renewed headphones
itemSearchSold(153142991698)
```

```
'4,194 sold'
```

```
##Williams Sonoma Americana Patriotic Stars Silicone Spatula Red White Blue Flag
itemSearchSold(254020780084)
```

```
'1 sold'
```



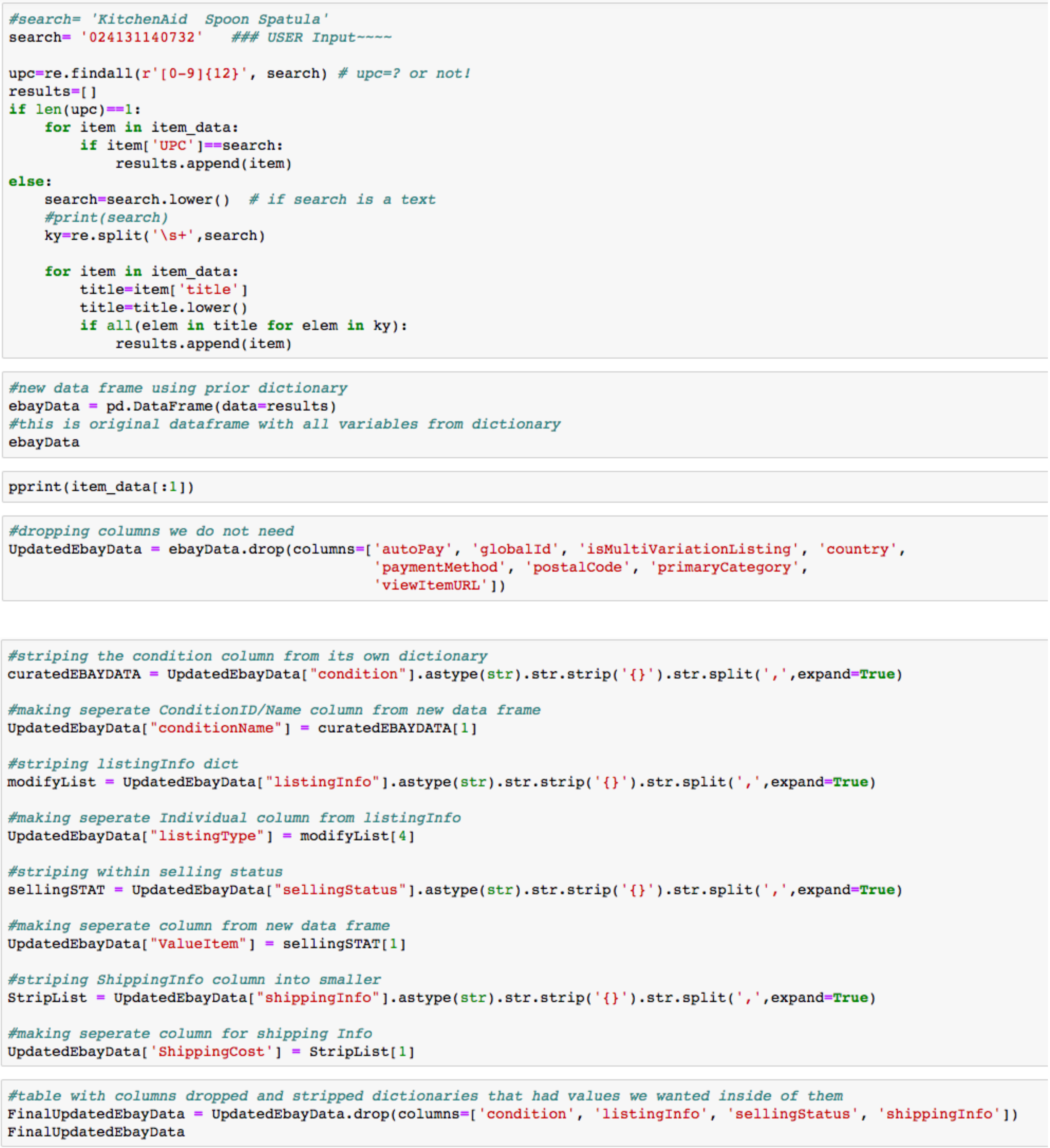
```
appendingList_updated=[]  # new updated list includes updated items
for item in appendingList:
#for item in appendingList[:5]:
    itemNumber=int(item['itemId'])
    item['UPC']=itemSearchUPC(itemNumber)             # update UPC
    item['Quantity']=itemSearchQuantity(itemNumber)   # update Quantity
    item['Sold']=itemSearchSold(itemNumber)           # update Sold
    appendingList_updated.append(item)
```

```
pprint(appendingList_updated)

{'Quantity': '10 available',
  'Sold': '581 sold',
  'UPC': '017817744188',
  'autoPay': 'true',
  'condition': {'conditionDisplayName': 'New', 'conditionId': '1000'},
  'country': 'US',
  'galleryURL': 'http://thumbs2.ebaystatic.com/m/mAtPtm4s044eVSDqh_6Ov-A/140.jpg',
  'globalId': 'EBAY-US',
  'isMultiVariationListing': 'false',
  'itemId': '142349176285',
  'listingInfo': {'bestOfferEnabled': 'false',
                  'buyItNowAvailable': 'false',
                  'endTime': '2019-01-04T19:42:16.000Z',
                  'gift': 'false',
                  'listingType': 'FixedPrice',
                  'startTime': '2017-04-14T19:42:16.000Z',
                  'watchCount': '555'},
  'location': 'New York,NY,USA',
  'paymentMethod': 'PayPal',
  'postalCode': '10040',
  'primaryCategory': {'categoryId': '111694',
                      'categoryName': 'Audio Docks & Mini Speakers'},
  'productId': {'_type': 'ReferenceID', 'value': '17019367610'},
  'returnsAccepted': 'true',
  'sellingStatus': {'convertedCurrentPrice': {'_currencyId': 'USD',
                                              'value': '179.0'},
                    'currentPrice': {'_currencyId': 'USD', 'value': '179.0'},
                    'sellingState': 'Active',
                    'timeLeft': 'P23DT22H13M56S'},
  'shippingInfo': {'expeditedShipping': 'true',
                   'handlingTime': '1',
                   'oneDayShippingAvailable': 'true',
                   'shipToLocations': 'Worldwide',
                   'shippingServiceCost': {'_currencyId': 'USD',
                                           'value': '0.0'},
                   'shippingType': 'Free'},
  'subtitle': '$9 GIFT CARD! AUTHORIZED DEALER! FREE 2 DAY SHIPPING!',
  'title': 'Bose SoundLink Revolve Bluetooth Speaker, Triple Black',
  'topRatedListing': 'false',
  'viewItemURL': 'http://www.ebay.com/itm/Bose-SoundLink-Revolve-Bluetooth-Speaker-Triple-Black-/142349176285'}
```

```
with open("data/mini_ebay_json_data_2.json", "w",encoding='utf-8') as f:
    # encoding='utf-8' for my computer (windows).  it may not ned mot Mac
    JSON_data = json.dumps(appendingList)
    f.write(JSON_data)
```

Search on Ebay Seller Estimator Program

```
from pprint import pprint
import csv
import json
import re
import pandas as pd

with open('mini_ebay_json_dataSET_2.json','r') as f:
    JSON_data=f.read()
    item_data=json.loads(JSON_data)
```



```
#search= 'KitchenAid  Spoon Spatula'
search= '024131140732'   ### USER Input~~~~

upc=re.findall(r'[0-9]{12}', search) # upc=? or not!
results=[]
if len(upc)==1:
    for item in item_data:
        if item['UPC']==search:
            results.append(item)
else:
    search=search.lower()  # if search is a text
    #print(search)
    ky=re.split('\s+',search)

    for item in item_data:
        title=item['title']
        title=title.lower()
        if all(elem in title for elem in ky):
            results.append(item)
```

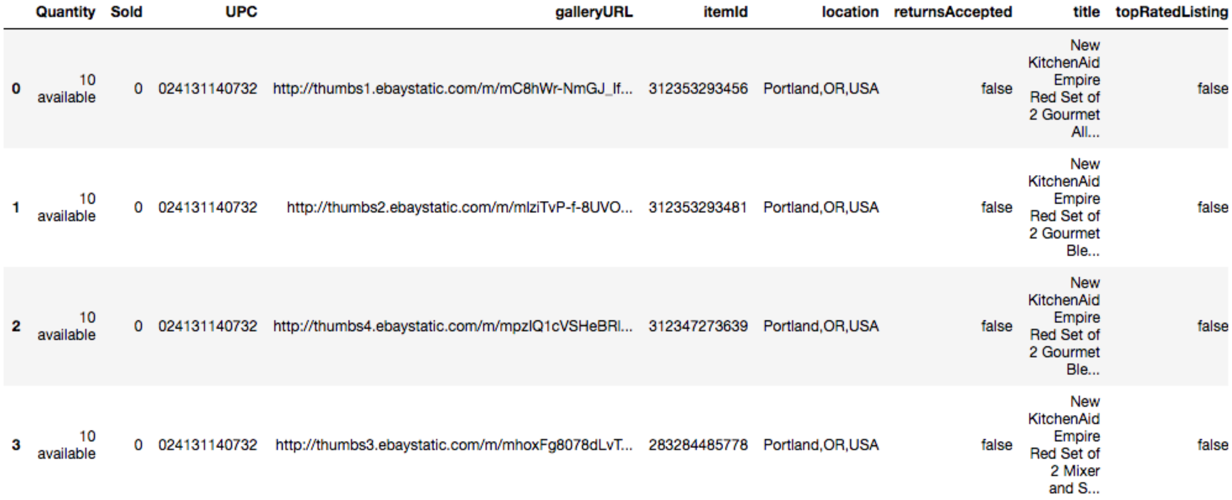
```
#new data frame using prior dictionary
ebayData = pd.DataFrame(data=results)
#this is original dataframe with all variables from dictionary
ebayData
```

```
pprint(item_data[:1])
```

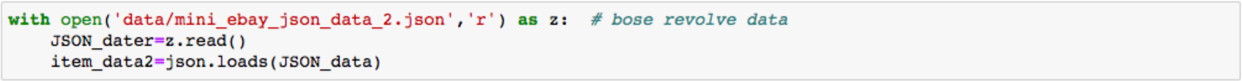
```
#dropping columns we do not need
UpdatedEbayData = ebayData.drop(columns=['autoPay', 'globalId', 'isMultiVariationListing', 'country',
                                         'paymentMethod', 'postalCode', 'primaryCategory',
                                         'viewItemURL'])
```

```
#striping the condition column from its own dictionary
curatedEBAYDATA = UpdatedEbayData["condition"].astype(str).str.strip('{}').str.split(',',expand=True)

#making seperate ConditionID/Name column from new data frame
UpdatedEbayData["conditionName"] = curatedEBAYDATA[1]

#striping listingInfo dict
modifyList = UpdatedEbayData["listingInfo"].astype(str).str.strip('{}').str.split(',',expand=True)

#making seperate Individual column from listingInfo
UpdatedEbayData["listingType"] = modifyList[4]

#striping within selling status
sellingSTAT = UpdatedEbayData["sellingStatus"].astype(str).str.strip('{}').str.split(',',expand=True)

#making seperate column from new data frame
UpdatedEbayData["ValueItem"] = sellingSTAT[1]

#striping ShippingInfo column into smaller
StripList = UpdatedEbayData["shippingInfo"].astype(str).str.strip('{}').str.split(',',expand=True)

#making seperate column for shipping Info
UpdatedEbayData['ShippingCost'] = StripList[1]
```

```
#table with columns dropped and stripped dictionaries that had values we wanted inside of them
FinalUpdatedEbayData = UpdatedEbayData.drop(columns=['condition', 'listingInfo', 'sellingStatus', 'shippingInfo'])
FinalUpdatedEbayData
```

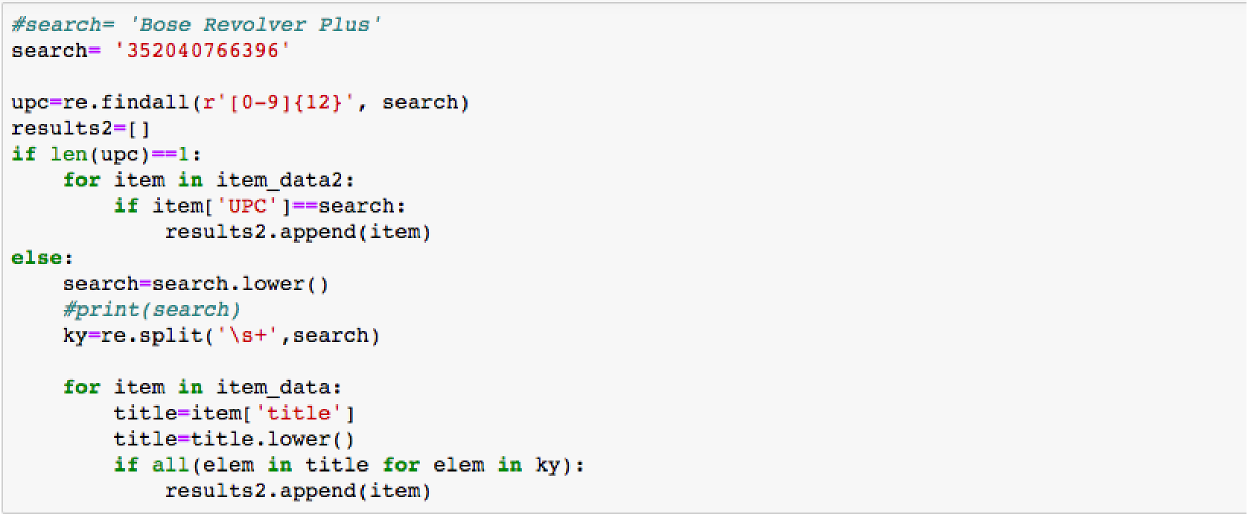


	Quantity	Sold	UPC	galleryURL	itemId	location	returnsAccepted	title	topRatedListing
0	10 available	0	024131140732	http://thumbs1.ebaystatic.com/m/mC8hWr-NmGJ_lf...	312353293456	Portland,OR,USA	false	New KitchenAid Empire Red Set of 2 Gourmet All...	false
1	10 available	0	024131140732	http://thumbs2.ebaystatic.com/m/mlziTvP-f-8UVO...	312353293481	Portland,OR,USA	false	New KitchenAid Empire Red Set of 2 Gourmet Ble...	false
2	10 available	0	024131140732	http://thumbs4.ebaystatic.com/m/mpzlQ1cVSHeBRl...	312347273639	Portland,OR,USA	false	New KitchenAid Empire Red Set of 2 Gourmet Ble...	false
3	10 available	0	024131140732	http://thumbs3.ebaystatic.com/m/mhoxFg8078dLvT...	283284485778	Portland,OR,USA	false	New KitchenAid Empire Red Set of 2 Mixer and S...	false

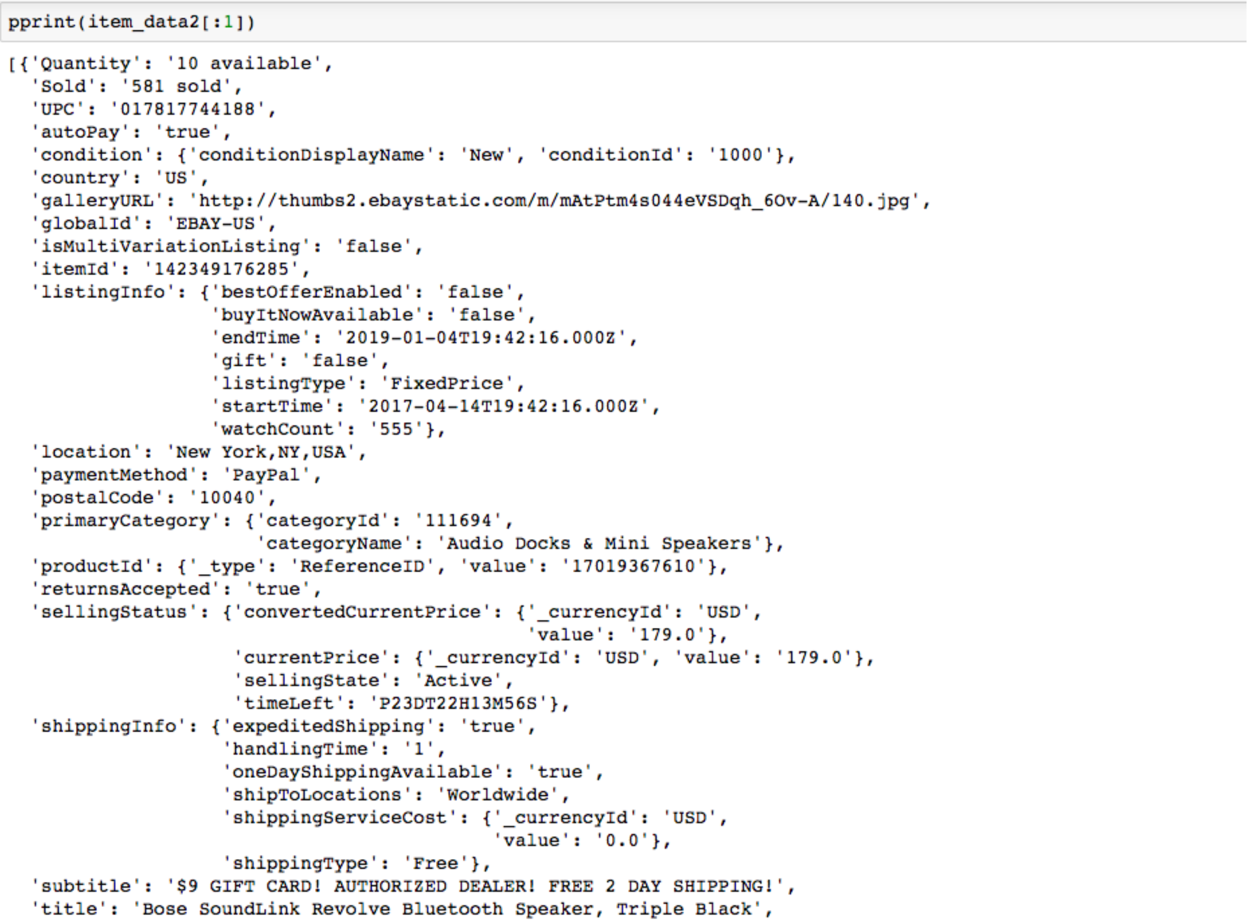


```
with open('data/mini_ebay_json_data_2.json','r') as z:  # bose revolve data
    JSON_dater=z.read()
    item_data2=json.loads(JSON_data)
```

Here below we implement our web scraping skills in finding information on the item number alone.



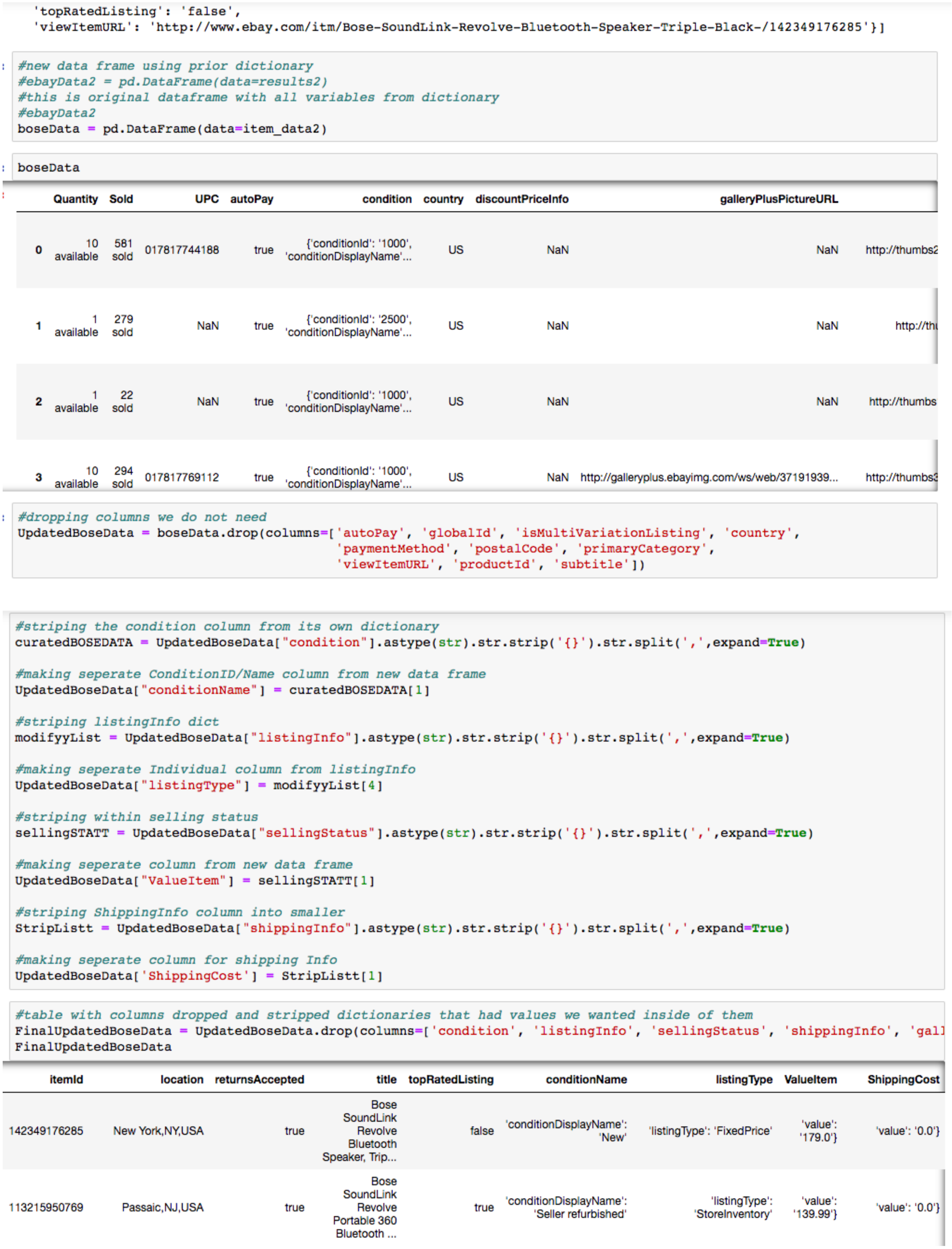
```
#search= 'Bose Revolver Plus'
search= '352040766396'

upc=re.findall(r'[0-9]{12}', search)
results2=[]
if len(upc)==1:
    for item in item_data2:
        if item['UPC']==search:
            results2.append(item)
else:
    search=search.lower()
    #print(search)
    ky=re.split('\s+',search)

    for item in item_data:
        title=item['title']
        title=title.lower()
        if all(elem in title for elem in ky):
            results2.append(item)
```



```
pprint(item_data2[:1])
```

```
[{'Quantity': '10 available',
  'Sold': '581 sold',
  'UPC': '017817744188',
  'autoPay': 'true',
  'condition': {'conditionDisplayName': 'New', 'conditionId': '1000'},
  'country': 'US',
  'galleryURL': 'http://thumbs2.ebaystatic.com/m/mAtPtm4s044eVSDqh_6Ov-A/140.jpg',
  'globalId': 'EBAY-US',
  'isMultiVariationListing': 'false',
  'itemId': '142349176285',
  'listingInfo': {'bestOfferEnabled': 'false',
                  'buyItNowAvailable': 'false',
                  'endTime': '2019-01-04T19:42:16.000Z',
                  'gift': 'false',
                  'listingType': 'FixedPrice',
                  'startTime': '2017-04-14T19:42:16.000Z',
                  'watchCount': '555'},
  'location': 'New York,NY,USA',
  'paymentMethod': 'PayPal',
  'postalCode': '10040',
  'primaryCategory': {'categoryId': '111694',
                      'categoryName': 'Audio Docks & Mini Speakers'},
  'productId': {'_type': 'ReferenceID', 'value': '17019367610'},
  'returnsAccepted': 'true',
  'sellingStatus': {'convertedCurrentPrice': {'_currencyId': 'USD',
                                              'value': '179.0'},
                    'currentPrice': {'_currencyId': 'USD', 'value': '179.0'},
                    'sellingState': 'Active',
                    'timeLeft': 'P23DT22H13M56S'},
  'shippingInfo': {'expeditedShipping': 'true',
                   'handlingTime': '1',
                   'oneDayShippingAvailable': 'true',
                   'shipToLocations': 'Worldwide',
                   'shippingServiceCost': {'_currencyId': 'USD',
                                           'value': '0.0'},
                   'shippingType': 'Free'},
  'subtitle': '$9 GIFT CARD! AUTHORIZED DEALER! FREE 2 DAY SHIPPING!',
  'title': 'Bose SoundLink Revolve Bluetooth Speaker, Triple Black',
```

Below we created a data frame that better showcases the dictionary.



```
 'topRatedListing': 'false',
 'viewItemURL': 'http://www.ebay.com/itm/Bose-SoundLink-Revolve-Bluetooth-Speaker-Triple-Black-/142349176285'}]
```

```
#new data frame using prior dictionary
#ebayData2 = pd.DataFrame(data=results2)
#this is original dataframe with all variables from dictionary
#ebayData2
boseData = pd.DataFrame(data=item_data2)
```

```
boseData
```

	Quantity	Sold	UPC	autoPay	condition	country	discountPriceInfo	galleryPlusPictureURL	
0	10 available	581 sold	017817744188	true	{'conditionId': '1000', 'conditionDisplayName'...	US	NaN	NaN	http://thumbs2
1	1 available	279 sold	NaN	true	{'conditionId': '2500', 'conditionDisplayName'...	US	NaN	NaN	http://thı
2	1 available	22 sold	NaN	true	{'conditionId': '1000', 'conditionDisplayName'...	US	NaN	NaN	http://thumbs
3	10 available	294 sold	017817769112	true	{'conditionId': '1000', 'conditionDisplayName'...	US	NaN	http://galleryplus.ebayimg.com/ws/web/37191939...	http://thumbs3

```
#dropping columns we do not need
UpdatedBoseData = boseData.drop(columns=['autoPay', 'globalId', 'isMultiVariationListing', 'country',
                                          'paymentMethod', 'postalCode', 'primaryCategory',
                                          'viewItemURL', 'productId', 'subtitle'])
```

```
#striping the condition column from its own dictionary
curatedBOSEDATA = UpdatedBoseData["condition"].astype(str).str.strip('{}').str.split(',',expand=True)

#making seperate ConditionID/Name column from new data frame
UpdatedBoseData["conditionName"] = curatedBOSEDATA[1]

#striping listingInfo dict
modifyyList = UpdatedBoseData["listingInfo"].astype(str).str.strip('{}').str.split(',',expand=True)

#making seperate Individual column from listingInfo
UpdatedBoseData["listingType"] = modifyyList[4]

#striping within selling status
sellingSTATT = UpdatedBoseData["sellingStatus"].astype(str).str.strip('{}').str.split(',',expand=True)

#making seperate column from new data frame
UpdatedBoseData["ValueItem"] = sellingSTATT[1]

#striping ShippingInfo column into smaller
StripListt = UpdatedBoseData["shippingInfo"].astype(str).str.strip('{}').str.split(',',expand=True)

#making seperate column for shipping Info
UpdatedBoseData['ShippingCost'] = StripListt[1]
```

```
#table with columns dropped and stripped dictionaries that had values we wanted inside of them
FinalUpdatedBoseData = UpdatedBoseData.drop(columns=['condition', 'listingInfo', 'sellingStatus', 'shippingInfo', 'gall
FinalUpdatedBoseData
```

itemId	location	returnsAccepted	title	topRatedListing	conditionName	listingType	ValueItem	ShippingCost
142349176285	New York,NY,USA	true	Bose SoundLink Revolve Bluetooth Speaker, Trip...	false	'conditionDisplayName': 'New'	'listingType': 'FixedPrice'	'value': '179.0'}	'value': '0.0'}
113215950769	Passaic,NJ,USA	true	Bose SoundLink Revolve Portable 360 Bluetooth ...	true	'conditionDisplayName': 'Seller refurbished'	'listingType': 'StoreInventory'	'value': '139.99'}	'value': '0.0'}