

**Name: Urinov Azizbek 20162672**

**Date: 10/12/2019**

**Class: Internet and Information Communication**

*Code explanation of non-chrome edition(BS).*

Codes	Code_Explanation
<pre>import requests from bs4 import BeautifulSoup</pre>	First import request command and then import BS from BS4.
<pre>headers = {'User-Agent': 'Chrome/66.0.3359.181'} url="http://jolse.com/category/sheet-masks/1027/"</pre>	“Header” command used to lead which server of chrome I want to use as a User-agent and “url” command is used to show PATH which website i wanna crawl.
<pre>result = requests.get(url, headers=headers) bs_obj = BeautifulSoup(result.content, "html.parser") print(bs_obj)</pre>	“result “ command take request and lead to PATH or URL. Then using with BS “bs_obj” Show all contents of “ <a href="http://jolse.com/category/sheet-masks/1027/">http://jolse.com/category/sheet-masks/1027/</a> ” website.
<pre>ul = bs_obj.find("ul", {"class": "prdList grid4"}) print(ul)</pre>	Go to “ul”(unordered list) and inside ul there is “class” named “prdList grid4” and show all results of “prdList grid4” under the name “ul”
<pre>boxes=ul.findAll("div", {"class": "box"}) for box in boxes:     print (box)</pre>	Go to >ul(inside ul)>div(inside div)>”class” named “box” and then print all results of “box”
<pre>for box in boxes:     ptag=box.find("p", {"class": "name"})     print(ptag)</pre>	Go to > “p”(inside “Box”)>class named “name” and save them all results to “ptag” and print “ptag”
<pre>for box in boxes:     ptag = box.find("p", {"class": "name"})     span = ptag.find("span")     print(span.text)</pre>	Go to > box and find “p” and “p” has “class” named as “name”. Inside “name” class there is “span” command which holds product names of website and show

	them as a text format.
<pre>def get_product_info(box):     ptag=box.find("p",{"class":"name"})     spans=ptag.find("span")     return spans.text for box in boxes:     product_info =get_product_info(box)     print(product_info)  for box in boxes:     price_tag = box.find("ul",{"class":"xans-product"})     price = price_tag.text     print (price)</pre>	<p>After taking the name of products inside &lt;span&gt; . Also scan product_info and print it.</p> <p>Create “price_tag” which holds product prices. First inside box find “p”&gt;”class”&gt;”xans-product”(holds product discount and real price). And print them all.</p>
<pre>def get_product_info(box):     ptag=box.find("p",{"class":"name"})     name_span=ptag.find("span")     price_ul=box.find("ul")     price_span=price_ul.findAll("span")      name=name_span.text     org_price=price_span[1].text     if len(price_span)&lt;4:         dis_price="No discount"      else:         dis_price=price_span[4].text      return {"name":name, "original_price":org_price, "discounted_price":dis_price,}  for box in boxes:     product_info = get_product_info(box)     print(product_info)</pre>	<p>Define one “attribute called get_product_info” which holds product_name, product_price and product_discount_price and save them all as a text format and show (by return option).</p> <p>-while scanning product, there are some cases in which some products don’t have Discount. In this case, “if/else” option used to just skip products which haven’t discount price and just show with own price.</p>
<pre>def get_page_products(url):     result = requests.get(url, headers=headers)     bs_obj = BeautifulSoup(result.content, "html.parser")     ul=bs_obj.find("ul", {"class": "prdList grid4"})     boxes=ul.findAll("div",{"class":"box"})     product_info_list = [get_product_info(box) for box in boxes]</pre>	<p>Sometimes we have to crawl not only one website page, we have to scan multiple pages and take information about products name, price and discount_price. In this case, first we have to call all items inside “box” and return to product_info_list.</p>

<pre> return product_info_list  urls=[  "http://jolse.com/category/sheet-masks/1027/?page=1" ,  "http://jolse.com/category/sheet-masks/1027/?page=2" ,  "http://jolse.com/category/sheet-masks/1027/?page=3" ,  "http://jolse.com/category/sheet-masks/1027/?page=4" ,  "http://jolse.com/category/sheet-masks/1027/?page=5" ,  "http://jolse.com/category/sheet-masks/1027/?page=6" ,  "http://jolse.com/category/sheet-masks/1027/?page=7" ,  "http://jolse.com/category/sheet-masks/1027/?page=8" ,  "http://jolse.com/category/sheet-masks/1027/?page=9" ,  "http://jolse.com/category/sheet-masks/1027/?page=10" ,  "http://jolse.com/category/sheet-masks/1027/?page=11" " ]  for page_number in range(0,11):     page_products =     get_page_products(urls[page_number])     print(len(page_products), page_products) </pre>	<p>Then, I have to insert all pages links(PATHs) as "url" (in my case they are 11pages) and then use for loop to scan each 11 pages and get information about price, name and discount_price(and print them all).</p>
--	---