```
#Importing required libraries
from bs4 import BeautifulSoup
import requests
import pandas as pd
from IPython.display import Image
```

```
# header = ['id', 'company_name', 'rating', 'reviews_count', 'address', 'category', 'phone', 'website']
source_data_url = 'https://www.amazon.in/s?k=iphones'
page = requests.get(source_data_url)
soup = BeautifulSoup(page.text, 'html')
soup.prettify()
```

```
'<!DOCTYPE html>\n<html class="a-no-js" data-19ax5a9jf="dingo" lang="en-in">\n <!-- sp:feature:head-start -->\n <head>\n  <script>\n
var aPageStart = (new Date()).getTime();\n  </script>\n  <meta charset="utf-8"/>\n  <!-- sp:end-feature:head-start -->\n  <!-- sp:featu
re:csm:head-open-part1 -->\n  <!-- sp:end-feature:csm:head-open-part1 -->\n  <!-- sp:feature:cs-optimization -->\n  <meta content="on"
http-equiv="x-dns-prefetch-control"/>\n  <link crossorigin="" href="https://images-eu.ssl-images-amazon.com" rel="dns-prefetch"/>\n  <l
ink crossorigin="" href="https://images-eu.ssl-images-amazon.com" rel="preconnect"/>\n  <link crossorigin="" href="https://m.media-amaz
on.com" rel="dns-prefetch"/>\n  <link crossorigin="" href="https://m.media-amazon.com" rel="preconnect"/>\n  <link crossorigin="" href
="https://completion.amazon.com" rel="dns-prefetch"/>\n  <link crossorigin="" href="https://completion.amazon.com" rel="preconnect"/>\n
<!-- sp:end-feature:cs-optimization -->\n  <!-- sp:feat  '
```

New Section

```
def appending_function(column):
    column_list=[]
    for i in range(0,len(column)):
        result= column[i].get_text()
        column_list.append(result)
    return  column_list
```

```
Title=soup.find_all('span', attrs={'class':'a-size-medium a-color-base a-text-normal'})
```

```
Title
appending_function(Title)
```

```
['Apple iPhone 13 (128GB) - Blue',
 'Apple iPhone 15 Plus (128 GB) - Green',
 'Apple iPhone 15 (128 GB) - Blue',
 'Apple iPhone 15 Plus (128 GB) - Black',
 'Apple iPhone 15 (256 GB) - Black',
 'Apple iPhone 15 (512 GB) - Pink',
 'Apple iPhone 15 Plus (256 GB) - Blue',
 'Apple iPhone 14 (256 GB) - (Product) RED',
 'Apple iPhone 15 Pro (256 GB) - Natural Titanium',
 'Apple iPhone 15 (256 GB) - Blue',
 'Apple iPhone 13 (128GB) - Pink',
 'Apple iPhone 15 Pro Max (256 GB) - Blue Titanium',
 'Apple iPhone 13 (128GB) - (Product) RED',
 'Apple iPhone 15 Plus (256 GB) - Black',
 'Apple iPhone 15 (128 GB) - Green',
 'Apple iPhone 15 Pro (128 GB) - Blue Titanium']
```

```
lastmonth_purchases= soup.find_all('span', attrs={'class':"a-size-base a-color-secondary"})
lastmonth_purchases[0].get_text().split()[0]
lastmonth_purchases_list=appending_function(lastmonth_purchases)
lastmonth_purchases_list1=[]
for i in range(len(lastmonth_purchases_list)-1):
     if  "bought in past month" in lastmonth_purchases_list[i]  and "M.R.P:" in  lastmonth_purchases_list[i+1]:
         lastmonth_purchases_list1.append(lastmonth_purchases_list[i].split()[0])
     elif "M.R.P" in lastmonth_purchases_list[i]  and "M.R.P:" in  lastmonth_purchases_list[i+1]:
          lastmonth_purchases_list1.append(" ")
lastmonth_purchases_list1
```

```
['3K+',
 '50+',
 '500+',
 '200+',
 '200+',
 ' ',
 '100+',
 ' ',
 '100+',
      '200+',
      '4K+',
      '200+',
      '1K+',
      '100+',
      ' ',
      '50+']
```

```
before_discount_price=soup.find_all('span', attrs={'class':"a-offscreen",'class':'a-text-price'})
#before_discount_price#[0].get_text().split("₹")[1]
before_discount_price=appending_function(before_discount_price)
before_discount_price=[x.split("₹")[1]  for x in before_discount_price]
before_discount_price
```

```
     ['59,900',
      '89,900',
      '79,900',
      '89,900',
      '89,900',
      '1,09,900',
      '99,900',
      '79,900',
      '1,44,900',
      '89,900',
      '59,900',
      '1,59,900',
      '59,900',
      '99,900',
      '79,990',
      '1,34,900']
```

```
After_discount_price=soup.find_all('span', attrs={'class':"a-offscreen",'class':'a-price'})
After_discount_price=appending_function(After_discount_price)
After_discount_price=After_discount_price[0::2]
After_discount_price=[x.split("₹")[1]  for x in After_discount_price]
```

```
len(After_discount_price)
```

```
     16
```

```
# After_discount_price_list = []
# Before_discount_price_list=[]
# for i in range(0,len(After_discount_price)):
#     result1= After_discount_price[i].get_text().split("₹")[1]
#     result2=before_discount_price[i].get_text().split("₹")[1]
#     After_discount_price_list.append(result1)
#     Before_discount_price_list.append(result2)
```

```
# lastmonth_purchases_list=[]
# for i in range(0,len(lastmonth_purchases)):
#       result3= lastmonth_purchases[i].get_text()
#       lastmonth_purchases_list.append(result3)
# # lastmonth_purchases_list

# # for x in lastmonth_purchases_list:
# #   if x != "M.R.P: ":
# #     lastmonth_purchases_list1.append(x.split()[0])
```

```
# #[0].get_text()#.split()[0]
```

```
dict = {'Title': Title,'Last_moth_purchase': lastmonth_purchases_list1,'Before_discount_price': before_discount_price,"After_discount_price"

df = pd.DataFrame(dict)
```

```
df
```

	Title	Last_moth_purchase	Before_discount_price	After_discount_price
0	[Apple iPhone 13 (128GB) - Blue]	3K+	59,900	52,999
1	[Apple iPhone 15 Plus (128 GB) - Green]	50+	89,900	80,990
2	[Apple iPhone 15 (128 GB) - Blue]	500+	79,900	71,990
3	[Apple iPhone 15 Plus (128 GB) - Black]	200+	89,900	80,990
4	[Apple iPhone 15 (256 GB) - Black]	200+	89,900	80,490
5	[Apple iPhone 15 (512 GB) - Pink]		1,09,900	99,990
6	[Apple iPhone 15 Plus (256 GB) - Blue]	100+	99,900	92,900
7	[Apple iPhone 14 (256 GB) - (Product)		79,900	65,998

```
# exporting data frame into csv/excel file
df.to_csv(r'D:\Python_project\Amazon.csv', index = False)
```

Start coding or generate with AI.