In [55]:

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

In [56]:

```
# Defining the URL of the website to scrape
url = "https://webscraper.io/test-sites/e-commerce/allinone/computers/laptop

# Sending a GET request to fetch the webpage content
r = requests.get(url)

# Parsing the webpage content using BeautifulSoup
soup  = BeautifulSoup(r.text, "html.parser")
```

In [57]:

```
#  Finding all 'a' tags with the class 'title' (containing product names)
product = soup.find_all("a", class_ = "title")

# Creating an empty list to store product names
products = []

# Looping through each product element and extracting the text
# and appending the product to products list
for p in product:
  products.append(p.text.strip('.'))

# Printing the total number of products found
print(len(products))
```

```
117
```

In [58]:

```
# Extracting all product prices from the webpage,
# storing them in a list, and printing the total count.
price = soup.find_all("h4", class_ = "price float-end card-title pull-right"
prices = []
for x in price:
  prices.append(x.text)
print(len(prices))
```

```
117
```

In [59]:

```
# Extracting all review counts from the webpage,
# storing them in a list, and printing the total count.
review = soup.find_all("p", class_ = "review-count float-end")
reviews = []
for r in review:
  reviews.append(r.text.replace('reviews', ''))
print(len(reviews))
```

```
117
```

In [60]:

```
# Extracting all 'data-rating' values from <p> tags and storing them in a li
rating = soup.select('p[data-rating]')
star_numbers = []
for r in rating:
    star_numbers.append(r.get('data-rating'))
len(star_numbers)
```

Out[60]: 117

In [61]:
```
# Finding all <p> tags with the classes description and card-text,
# Extracting their text content, and storing it in a list.
description = soup.find_all("p", class_ = "description card-text")
descriptions = []
for d in description:
  descriptions.append(d.text)
len(descriptions)
```

Out[61]: 117

In [62]:
```
# Combining the lists into a data frame using zip
df = pd.DataFrame(zip(products, prices, reviews, star_numbers, descriptions)
                  columns = ['Product', 'Price', 'Number of Reviews', 'Stars
df.head()
```

Out[62]:

	Product	Price	Number of Reviews	Stars	Description
0	Asus VivoBook	$295.99	14	3	Asus VivoBook X441NA-GA190 Chocolate Black, 14...
1	Prestigio Smar	$299	8	2	Prestigio SmartBook 133S Dark Grey, 13.3" FHD ...
2	Prestigio Smar	$299	12	4	Prestigio SmartBook 133S Gold, 13.3" FHD IPS, ...
3	Aspire E1-510	$306.99	2	3	15.6", Pentium N3520 2.16GHz, 4GB, 500GB, Linux
4	Lenovo V110-15	$321.94	5	3	Lenovo V110-15IAP, 15.6" HD, Celeron N3350 1.1...

In [62]: