

# cars24 Web Scraping

August 23, 2024

```
[3]: !pip install requests beautifulsoup4 pandas
import requests
from bs4 import BeautifulSoup
import pandas as pd
# URL of the Cars24 page you want to scrape
url = 'https://www.cars24.com/buy-used-car?
↳sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
↳1724220956&storeCityId=2378'

# Send a GET request to fetch the HTML content
response = requests.get(url)

# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(response.content, 'html.parser')

# Initialize lists to store data
car_names = []
car_prices = []
car_kms = []
car_locations = []

# Find all car listings using the structure provided
cars = soup.find_all('div', class_='_2YB7p')

# Loop through each car listing and extract details
for car in cars:
    name = car.find('h3', class='_11dVb').text if car.find('h3',
↳class='_11dVb') else 'N/A'
    price = car.find('div', class='_2KyOK').text if car.find('div',
↳class='_2KyOK') else 'N/A'

    # Extract kilometers driven from `ul` tag with class `_3J2G-`
    km_tag = car.find('ul', class='_3J2G-')
    km_driven = km_tag.text.strip() if km_tag else 'N/A'

    # Extract location information from `p` with class `_3dGMY`
    location_tag = car.find('p', class='_3dGMY')
```

```

        location_text = ' '.join(span.text for span in location_tag.
↪find_all('span')) if location_tag else 'N/A'

        car_names.append(name)
        car_prices.append(price)
        car_kms.append(km_driven)
        car_locations.append(location_text)

# Create a DataFrame from the extracted data
df = pd.DataFrame({
    'Car Name': car_names,
    'Price': car_prices,
    'Kilometers Driven': car_kms,
    'Location': car_locations
})

df

```

Requirement already satisfied: requests in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(2.32.3)

Requirement already satisfied: beautifulsoup4 in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(4.12.3)

Requirement already satisfied: pandas in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(2.2.2)

Requirement already satisfied: charset-normalizer<4,>=2 in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(from requests) (3.3.2)

Requirement already satisfied: idna<4,>=2.5 in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(from requests) (3.7)

Requirement already satisfied: urllib3<3,>=1.21.1 in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(from requests) (1.26.19)

Requirement already satisfied: certifi>=2017.4.17 in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(from requests) (2024.7.4)

Requirement already satisfied: soupsieve>1.2 in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(from beautifulsoup4) (2.5)

Requirement already satisfied: numpy>=1.26.0 in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(from pandas) (2.0.1)

Requirement already satisfied: python-dateutil>=2.8.2 in  
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages  
(from pandas) (2.9.0.post0)

```
Requirement already satisfied: pytz>=2020.1 in
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages
(from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages
(from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in
c:\users\kongegowtam\appdata\local\programs\python\python312\lib\site-packages
(from python-dateutil>=2.8.2->pandas) (1.16.0)
```

[notice] A new release of pip is available: 24.1.2 -> 24.2

[notice] To update, run: python.exe -m pip install --upgrade pip

[3]:

	Car Name \	Price	Kilometers Driven \
0	2016 Honda BR-V 1.5L I-VTEC V	6.46 Lakh	76,235 kmPetrolManual
1	2015 Volkswagen Polo COMFORTLINE 1.2L	3.83 Lakh	66,024 kmPetrolManual
2	2016 Tata Tiago XZ PETROL	3.94 Lakh 4.25 Lakh(31.01k off)	33,323 kmPetrolManual
3	2016 Renault Kwid RXT 1.0	2.85 Lakh 3.03 Lakh(18k off)	32,258 kmPetrolManual
4	2020 Maruti Baleno DELTA PETROL 1.2	6.29 Lakh 7.16 Lakh(86.54k off)	35,304 kmPetrolManual
5	2018 Renault Kwid CLIMBER 1.0 AMT	3.71 Lakh 4.01 Lakh(30.09k off)	60,067 kmPetrolAutomatic
6	2022 KIA SELTOS GTX (0) 1.4	14.76 Lakh	23,040 kmPetrolManual
7	2016 Hyundai Creta SX PLUS AT 1.6 DIESEL	8.34 Lakh	33,417 kmDieselAutomatic
8	2023 Maruti XL6 ZETA CNG	12.28 Lakh	32,447 kmCNGManual
9	2013 Maruti Wagon R 1.0 LXi CNG	2.74 Lakh 3.12 Lakh(37.78k off)	64,400 kmCNGManual
10	2017 Honda WR-V 1.5L I-DTEC VX MT	7.02 Lakh	40,039 kmDieselManual
11	2017 Hyundai Elite i20 MAGNA EXECUTIVE 1.2		
12	2018 Hyundai i20 Active 1.2 SX		
13	2023 Tata TIGOR XZ PLUS CNG		
14	2023 Hyundai VENUE SX 1.5 CRDI		
15	2015 Hyundai i20 Active 1.2 S		
16	2021 Tata NEXON XE PETROL		
17	2020 KIA SONET GTX PLUS 1.0 IMT		
18	2019 Tata Harrier XZ 2.0L		
19	2014 Honda City 1.5L I-VTEC VX		

11	5.15 Lakh 6.06 Lakh(91.36k off)	31,359 kmPetrolManual
12	5.83 Lakh 6.91 Lakh(1.08L off)	85,880 kmPetrolManual
13	7.92 Lakh 9.00 Lakh(1.08L off)	3,817 kmCNGManual
14	11.38 Lakh	24,566 kmDieselManual
15	4.78 Lakh 5.68 Lakh(89.79k off)	53,664 kmPetrolManual
16	6.51 Lakh 7.60 Lakh(1.09L off)	27,603 kmPetrolManual
17	10.04 Lakh 10.34 Lakh(29.72k off)	35,415 kmPetrolManual
18	12.71 Lakh 12.84 Lakh(13k off)	57,037 kmDieselManual
19	5.31 Lakh 6.04 Lakh(73k off)	34,473 kmPetrolManual

	Location
0	Today Goregaon, Mumbai
1	Today Seawood\n
2	Today Mulund West, Mumbai \n
3	Today Mulund West, Mumbai \n
4	Today Goregaon, Mumbai
5	Today Goregaon, Mumbai
6	Today Goregaon, Mumbai
7	Today Goregaon, Mumbai
8	Today Goregaon, Mumbai
9	Today Mulund West, Mumbai \n
10	Today Seawood\n
11	Today Mulund West, Mumbai \n
12	Today Goregaon, Mumbai
13	Today Goregaon, Mumbai
14	Today Goregaon, Mumbai
15	Today Mulund West, Mumbai \n
16	Today Mulund West, Mumbai \n
17	Today Goregaon, Mumbai
18	Today Goregaon, Mumbai
19	Today Goregaon, Mumbai

```
[9]: import requests
from bs4 import BeautifulSoup
import pandas as pd

def get_page_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Initialize lists to store data
    car_names = []
    car_prices = []
    car_kms = []
    car_locations = []

    # Find all car listings using the structure provided
```

```

cars = soup.find_all('div', class_='_2YB7p')

# Loop through each car listing and extract details
for car in cars:
    name = car.find('h3', class_='_11dVb').text if car.find('h3', class_='_11dVb') else 'N/A'
    price = car.find('div', class_='_2KyOK').text if car.find('div', class_='_2KyOK') else 'N/A'

    # Extract kilometers driven from `ul` tag with class `_3J2G-`
    km_tag = car.find('ul', class_='_3J2G-')
    km_driven = km_tag.text.strip() if km_tag else 'N/A'

    # Extract location information from `p` with class `_3dGMY`
    location_tag = car.find('p', class_='_3dGMY')
    location_text = ' '.join(span.text for span in location_tag.find_all('span')) if location_tag else 'N/A'

    car_names.append(name)
    car_prices.append(price)
    car_kms.append(km_driven)
    car_locations.append(location_text)

return car_names, car_prices, car_kms, car_locations

base_url = 'https://www.cars24.com/buy-used-car?
    ↪sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
    ↪1724220956&storeCityId=2378&page='

all_car_names = []
all_car_prices = []
all_car_kms = []
all_car_locations = []

# Number of pages to scrape
num_pages = 10

for page in range(1, num_pages + 1):
    url = base_url + str(page)
    car_names, car_prices, car_kms, car_locations = get_page_data(url)
    all_car_names.extend(car_names)
    all_car_prices.extend(car_prices)
    all_car_kms.extend(car_kms)
    all_car_locations.extend(car_locations)

# Create a DataFrame from the extracted data
df = pd.DataFrame({

```

```

'Car Name': all_car_names,
'Price': all_car_prices,
'Kilometers Driven': all_car_kms,
'Location': all_car_locations
})

df

```

```

[9]:
      Car Name \
0      2016 Honda BR-V 1.5L I-VTEC V
1      2015 Volkswagen Polo COMFORTLINE 1.2L
2      2016 Tata Tiago XZ PETROL
3      2016 Renault Kwid RXT 1.0
4      2020 Maruti Baleno DELTA PETROL 1.2
..
195     2015 Hyundai i20 Active 1.2 S
196     2021 Tata NEXON XE PETROL
197     2020 KIA SONET GTX PLUS 1.0 IMT
198     2019 Tata Harrier XZ 2.0L
199     2014 Honda City 1.5L I-VTEC VX

      Price      Kilometers Driven \
0      6.46 Lakh  76,235 kmPetrolManual
1      3.83 Lakh  66,024 kmPetrolManual
2      3.94 Lakh 4.25 Lakh(31.01k off) 33,323 kmPetrolManual
3      2.85 Lakh 3.03 Lakh(18k off) 32,258 kmPetrolManual
4      6.29 Lakh 7.16 Lakh(86.54k off) 35,304 kmPetrolManual
..
195     4.78 Lakh 5.68 Lakh(89.79k off) 53,664 kmPetrolManual
196     6.51 Lakh 7.60 Lakh(1.09L off) 27,603 kmPetrolManual
197    10.04 Lakh 10.34 Lakh(29.72k off) 35,415 kmPetrolManual
198     12.71 Lakh 12.84 Lakh(13k off) 57,037 kmDieselManual
199     5.31 Lakh 6.04 Lakh(73k off) 34,473 kmPetrolManual

      Location
0      Today Goregaon, Mumbai
1      Today Seawood\n
2      Today Mulund West, Mumbai \n
3      Today Mulund West, Mumbai \n
4      Today Goregaon, Mumbai
..
195 Today Mulund West, Mumbai \n
196 Today Mulund West, Mumbai \n
197      Today Goregaon, Mumbai
198      Today Goregaon, Mumbai
199      Today Goregaon, Mumbai

```

[200 rows x 4 columns]

```
[12]: import requests
from bs4 import BeautifulSoup
import pandas as pd
from IPython.display import FileLink

def get_page_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Initialize lists to store data
    car_names = []
    car_prices = []
    car_kms = []
    car_locations = []

    # Find all car listings using the structure provided
    cars = soup.find_all('div', class_='_2YB7p')

    # Loop through each car listing and extract details
    for car in cars:
        name = car.find('h3', class_='_11dVb').text if car.find('h3',
↪class_='_11dVb') else 'N/A'
        price = car.find('div', class_='_2KyOK').text if car.find('div',
↪class_='_2KyOK') else 'N/A'

        # Extract kilometers driven from `ul` tag with class `_3J2G-`
        km_tag = car.find('ul', class_='_3J2G-')
        km_driven = km_tag.text.strip() if km_tag else 'N/A'

        # Extract location information from `p` with class `_3dGMY`
        location_tag = car.find('p', class_='_3dGMY')
        location_text = ' '.join(span.text for span in location_tag.
↪find_all('span')) if location_tag else 'N/A'

        car_names.append(name)
        car_prices.append(price)
        car_kms.append(km_driven)
        car_locations.append(location_text)

    return car_names, car_prices, car_kms, car_locations

base_url = 'https://www.cars24.com/buy-used-car?
↪sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
↪1724220956&storeCityId=2378&page='
```

```

all_car_names = []
all_car_prices = []
all_car_kms = []
all_car_locations = []

# Number of pages to scrape
num_pages = 10

for page in range(1, num_pages + 1):
    url = base_url + str(page)
    car_names, car_prices, car_kms, car_locations = get_page_data(url)
    all_car_names.extend(car_names)
    all_car_prices.extend(car_prices)
    all_car_kms.extend(car_kms)
    all_car_locations.extend(car_locations)

# Create a DataFrame from the extracted data
df = pd.DataFrame({
    'Car Name': all_car_names,
    'Price': all_car_prices,
    'Kilometers Driven': all_car_kms,
    'Location': all_car_locations
})

# Save the DataFrame to a CSV file
csv_filename = 'cars24_mumbai.csv'
df.to_csv(csv_filename, index=False)

# Create a downloadable link for the CSV file
FileLink(csv_filename)

```

[12]: C:\Users\KONGEGOWTAM\cars24\_mumbai.csv

```

[14]: import requests
from bs4 import BeautifulSoup
import pandas as pd
from IPython.display import FileLink

def get_page_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Initialize lists to store data
    car_names = []
    car_prices = []
    car_kms = []
    car_locations = []

```



```

# Find all car listings using the structure provided
cars = soup.find_all('div', class='_2YB7p')

# Loop through each car listing and extract details
for car in cars:
    name = car.find('h3', class='_11dVb').text if car.find('h3', class='_11dVb') else 'N/A'
    price = car.find('div', class='_2KyOK').text if car.find('div', class='_2KyOK') else 'N/A'

    # Extract kilometers driven from `ul` tag with class `_3J2G-`
    km_tag = car.find('ul', class='_3J2G-')
    km_driven = km_tag.text.strip() if km_tag else 'N/A'

    # Extract location information from `p` with class `_3dGMY`
    location_tag = car.find('p', class='_3dGMY')
    location_text = ' '.join(span.text for span in location_tag.find_all('span')) if location_tag else 'N/A'

    car_names.append(name)
    car_prices.append(price)
    car_kms.append(km_driven)
    car_locations.append(location_text)

return car_names, car_prices, car_kms, car_locations

base_url = 'https://www.cars24.com/buy-used-car?
sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
1724220956&storeCityId='

all_car_names = []
all_car_prices = []
all_car_kms = []
all_car_locations = []

# Number of pages to scrape
num_pages = 10

for page in range(1, num_pages + 1):
    url = base_url + str(page)
    car_names, car_prices, car_kms, car_locations = get_page_data(url)
    all_car_names.extend(car_names)
    all_car_prices.extend(car_prices)
    all_car_kms.extend(car_kms)
    all_car_locations.extend(car_locations)

```

```

# Create a DataFrame from the extracted data
df = pd.DataFrame({
    'Car Name': all_car_names,
    'Price': all_car_prices,
    'Kilometers Driven': all_car_kms,
    'Location': all_car_locations
})

# Save the DataFrame to a CSV file
csv_filename = 'cars24_New Delhi.csv'
df.to_csv(csv_filename, index=False)

# Create a downloadable link for the CSV file
FileLink(csv_filename)

```

[14]: C:\Users\KONGEGOWTAM\cars24\_New Delhi.csv

```

[40]: import requests
from bs4 import BeautifulSoup
import pandas as pd
from IPython.display import FileLink

def get_page_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Initialize lists to store data
    car_names = []
    car_prices = []
    car_kms = []
    car_locations = []

    # Find all car listings using the structure provided
    cars = soup.find_all('div', class_='_2YB7p')

    # Loop through each car listing and extract details
    for car in cars:
        name = car.find('h3', class_='_11dVb').text if car.find('h3', class_='_11dVb') else 'N/A'
        price = car.find('div', class_='_2KyOK').text if car.find('div', class_='_2KyOK') else 'N/A'

        # Extract kilometers driven from `ul` tag with class `_3J2G-`
        km_tag = car.find('ul', class_='_3J2G-')
        km_driven = km_tag.text.strip() if km_tag else 'N/A'

        # Extract location information from `p` with class `_3dGMY`

```

```

        location_tag = car.find('p',class='_3dGMY')
        location_text = ' '.join(span.text for span in location_tag.
↪find_all('span')) if location_tag else 'N/A'

        car_names.append(name)
        car_prices.append(price)
        car_kms.append(km_driven)
        car_locations.append(location_text)

    return car_names, car_prices, car_kms, car_locations

base_url = 'https://www.cars24.com/buy-used-car?
↪sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
↪1724220956&storeCityId=3686'

all_car_names = []
all_car_prices = []
all_car_kms = []
all_car_locations = []

# Number of pages to scrape
num_pages = 10

for page in range(1, num_pages + 1):
    url = base_url + str(page)
    car_names, car_prices, car_kms, car_locations = get_page_data(url)
    all_car_names.extend(car_names)
    all_car_prices.extend(car_prices)
    all_car_kms.extend(car_kms)
    all_car_locations.extend(car_locations)

# Create a DataFrame from the extracted data
df = pd.DataFrame({
    'Car Name': all_car_names,
    'Price': all_car_prices,
    'Kilometers Driven': all_car_kms,
    'Location': all_car_locations
})

df

```

[40]: Empty DataFrame  
Columns: [Car Name, Price, Kilometers Driven, Location]  
Index: []

```

[41]: import requests
from bs4 import BeautifulSoup
import pandas as pd
from IPython.display import FileLink

def get_page_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Initialize lists to store data
    car_names = []
    car_prices = []
    car_kms = []
    car_locations = []

    # Find all car listings using the structure provided
    cars = soup.find_all('div', class_='_2YB7p')

    print(f"Found {len(cars)} cars on the page.")

    # Loop through each car listing and extract details
    for car in cars:
        name = car.find('h3', class_='_11dVb').text if car.find('h3', class_='_11dVb') else 'N/A'
        price = car.find('div', class_='_2KyOK').text if car.find('div', class_='_2KyOK') else 'N/A'

        # Extract kilometers driven from `ul` tag with class `_3J2G-`
        km_tag = car.find('ul', class_='_3J2G-')
        km_driven = km_tag.text.strip() if km_tag else 'N/A'

        # Extract location information from `p` with class `_3dGMY`
        location_tag = car.find('p', class_='_3dGMY')
        location_text = ' '.join(span.text for span in location_tag.find_all('span')) if location_tag else 'N/A'

        car_names.append(name)
        car_prices.append(price)
        car_kms.append(km_driven)
        car_locations.append(location_text)

    return car_names, car_prices, car_kms, car_locations

base_url = 'https://www.cars24.com/buy-used-car?
↳sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
↳1724220956&storeCityId=3686&page='

```

```

all_car_names = []
all_car_prices = []
all_car_kms = []
all_car_locations = []

# Number of pages to scrape
num_pages = 10

for page in range(1, num_pages + 1):
    url = base_url + str(page)
    print(f"Scraping page {page} with URL: {url}")
    car_names, car_prices, car_kms, car_locations = get_page_data(url)
    all_car_names.extend(car_names)
    all_car_prices.extend(car_prices)
    all_car_kms.extend(car_kms)
    all_car_locations.extend(car_locations)

# Create a DataFrame from the extracted data
df = pd.DataFrame({
    'Car Name': all_car_names,
    'Price': all_car_prices,
    'Kilometers Driven': all_car_kms,
    'Location': all_car_locations
})

# Display the first few rows of the DataFrame
df.head()

```

Scraping page 1 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=1>  
Found 20 cars on the page.

Scraping page 2 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=2>  
Found 20 cars on the page.

Scraping page 3 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=3>  
Found 20 cars on the page.

Scraping page 4 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=4>  
Found 20 cars on the page.

Scraping page 5 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=5>  
Found 20 cars on the page.

Scraping page 6 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=6>  
Found 20 cars on the page.

Scraping page 7 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=7>

Found 20 cars on the page.

Scraping page 8 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=8>

Found 20 cars on the page.

Scraping page 9 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=9>

Found 20 cars on the page.

Scraping page 10 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=3686&page=10>

Found 20 cars on the page.

```
[41]:
```

	Car Name	Price \
0	2019 Volkswagen Polo GT TSI AT	7.41 Lakh
1	2021 Mahindra Thar LX HT PETROL 4WD MT	12.83 Lakh
2	2018 Tata Tiago XZA PETROL	4.85 Lakh
3	2017 Volkswagen Ameo HIGHLINE1.5L	5.76 Lakh
4	2022 Nissan MAGNITE XV TURBO CVT DUAL TONE	8.68 Lakh

	Kilometers Driven	Location
0	94,669 kmPetrolAutomatic	Today Kompally, Hyderabad
1	23,965 kmPetrolManual	Today Bachupally, Hyderabad
2	41,772 kmPetrolAutomatic	Today Bachupally, Hyderabad
3	1,08,064 kmDieselManual	Today Attapur, Hyderabad
4	13,925 kmPetrolAutomatic	Today Kompally, Hyderabad

```
[42]: df.tail()
```

```
[42]:
```

	Car Name \
195	2017 Maruti Swift VXI
196	2021 Renault Kwid CLIMBER 1.0 (0)
197	2019 Honda Amaze 1.2L I-VTEC VX
198	2016 Maruti Swift VDI
199	2018 Hyundai Grand i10 MAGNA 1.2 KAPPA VTVT

  

	Price	Kilometers Driven \
195	5.61 Lakh	25,847 kmPetrolManual
196	4.47 Lakh 5.36 Lakh(89.13k off)	7,391 kmPetrolManual
197	6.89 Lakh	37,615 kmPetrolManual
198	5.17 Lakh	61,992 kmDieselManual
199	4.44 Lakh 5.26 Lakh(81.9k off)	50,918 kmPetrolManual

  

	Location
195	Today Attapur, Hyderabad
196	Today Kompally, Hyderabad
197	Today Kompally, Hyderabad
198	Today Attapur, Hyderabad
199	Today Kompally, Hyderabad

```
[43]: df
```

```
[43]:
```

	Car Name \		Price	Kilometers Driven \
0	2019 Volkswagen Polo GT TSI AT		7.41 Lakh	94,669 kmPetrolAutomatic
1	2021 Mahindra Thar LX HT PETROL 4WD MT		12.83 Lakh	23,965 kmPetrolManual
2	2018 Tata Tiago XZA PETROL		4.85 Lakh	41,772 kmPetrolAutomatic
3	2017 Volkswagen Ameo HIGHLINE1.5L		5.76 Lakh	1,08,064 kmDieselManual
4	2022 Nissan MAGNITE XV TURBO CVT DUAL TONE		8.68 Lakh	13,925 kmPetrolAutomatic
..	...		...	...
195	2017 Maruti Swift VXi		5.61 Lakh	25,847 kmPetrolManual
196	2021 Renault Kwid CLIMBER 1.0 (0)	4.47 Lakh 5.36 Lakh(89.13k off)	7,391 kmPetrolManual	
197	2019 Honda Amaze 1.2L I-VTEC VX		6.89 Lakh	37,615 kmPetrolManual
198	2016 Maruti Swift VDI		5.17 Lakh	61,992 kmDieselManual
199	2018 Hyundai Grand i10 MAGNA 1.2 KAPPA VTVT	4.44 Lakh 5.26 Lakh(81.9k off)	50,918 kmPetrolManual	

  

	Location
0	Today Kompally, Hyderabad
1	Today Bachupally, Hyderabad
2	Today Bachupally, Hyderabad
3	Today Attapur, Hyderabad
4	Today Kompally, Hyderabad
..	...
195	Today Attapur, Hyderabad
196	Today Kompally, Hyderabad
197	Today Kompally, Hyderabad
198	Today Attapur, Hyderabad
199	Today Kompally, Hyderabad

```
[200 rows x 4 columns]
```

```
[44]: # Save the DataFrame to a CSV file
csv_filename = 'cars24_Hyderabad.csv'
df.to_csv(csv_filename, index=False)
```

```
# Create a downloadable link for the CSV file
FileLink(csv_filename)
```

[44]: C:\Users\KONGEGOWTAM\cars24\_Hyderabad.csv

```
[45]: import requests
from bs4 import BeautifulSoup
import pandas as pd
from IPython.display import FileLink

def get_page_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Initialize lists to store data
    car_names = []
    car_prices = []
    car_kms = []
    car_locations = []

    # Find all car listings using the structure provided
    cars = soup.find_all('div', class_='_2YB7p')

    print(f"Found {len(cars)} cars on the page.")

    # Loop through each car listing and extract details
    for car in cars:
        name = car.find('h3', class='_11dVb').text if car.find('h3', class='_11dVb') else 'N/A'
        price = car.find('div', class='_2KyOK').text if car.find('div', class='_2KyOK') else 'N/A'

        # Extract kilometers driven from `ul` tag with class `_3J2G-`
        km_tag = car.find('ul', class='_3J2G-')
        km_driven = km_tag.text.strip() if km_tag else 'N/A'

        # Extract location information from `p` with class `_3dGMY`
        location_tag = car.find('p', class='_3dGMY')
        location_text = ' '.join(span.text for span in location_tag.find_all('span')) if location_tag else 'N/A'

        car_names.append(name)
        car_prices.append(price)
        car_kms.append(km_driven)
        car_locations.append(location_text)

    return car_names, car_prices, car_kms, car_locations
```



```

base_url = 'https://www.cars24.com/buy-used-car?
↳sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
↳1724220956&storeCityId=4709&page='

all_car_names = []
all_car_prices = []
all_car_kms = []
all_car_locations = []

# Number of pages to scrape
num_pages = 10

for page in range(1, num_pages + 1):
    url = base_url + str(page)
    print(f"Scraping page {page} with URL: {url}")
    car_names, car_prices, car_kms, car_locations = get_page_data(url)
    all_car_names.extend(car_names)
    all_car_prices.extend(car_prices)
    all_car_kms.extend(car_kms)
    all_car_locations.extend(car_locations)

# Create a DataFrame from the extracted data
df = pd.DataFrame({
    'Car Name': all_car_names,
    'Price': all_car_prices,
    'Kilometers Driven': all_car_kms,
    'Location': all_car_locations
})
df
# Save DataFrame to CSV
csv_file = 'car_Bangalore.csv'
df.to_csv(csv_file, index=False)

# Provide a download link
FileLink(csv_file)

```

Scraping page 1 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=1>

Found 20 cars on the page.

Scraping page 2 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=2>

Found 20 cars on the page.

Scraping page 3 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=3>

Found 20 cars on the page.

Scraping page 4 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=4>

Found 20 cars on the page.  
 Scraping page 5 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=5>  
 Found 20 cars on the page.  
 Scraping page 6 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=6>  
 Found 20 cars on the page.  
 Scraping page 7 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=7>  
 Found 20 cars on the page.  
 Scraping page 8 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=8>  
 Found 20 cars on the page.  
 Scraping page 9 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=9>  
 Found 20 cars on the page.  
 Scraping page 10 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=4709&page=10>  
 Found 20 cars on the page.

[45]: C:\Users\KONGEGOWTAM\car\_Bangalore.csv

[46]: df

[46]:

	Car Name	Price \
0	2013 Honda City 1.5L I-VTEC V MT	4.64 Lakh
1	2015 Renault Duster 110 PS RXL DIESEL	5.74 Lakh
2	2021 Tata PUNCH ACCOMPLISHED AMT	8.49 Lakh
3	2016 Ford Ecosport TITANIUM 1.5L PETROL	6.16 Lakh
4	2017 Renault Duster 85 PS RXZ DIESEL	7.72 Lakh 9.12 Lakh(1.4L off)
..	...	...
195	2018 Skoda Rapid STYLE 1.6 MPI	7.12 Lakh
196	2018 Honda Jazz 1.2L I-VTEC V	5.57 Lakh
197	2015 Honda City 1.5L I-VTEC SV	6.08 Lakh
198	2017 Renault Kwid RXT 1.0 (0)	3.42 Lakh
199	2017 Maruti Baleno RS 1.0 PETROL	6.70 Lakh

  

	Kilometers Driven \
0	84,818 kmPetrolManual
1	94,510 kmDieselManual
2	8,945 kmPetrolAutomatic
3	71,617 kmPetrolManual
4	71,918 kmDieselManual
..	...
195	70,771 kmPetrolManual
196	31,348 kmPetrolManual
197	53,160 kmPetrolManual
198	12,883 kmPetrolManual

199      36,274 kmPetrolManual

			Location
0	Today	Whitefield Main Road, Bhoruka Tech Park...	
1		Today	Bellahalli, Bengaluru
2		Today	Garuda Mall, Magrath Road, Bengaluru
3		Today	Garuda Mall, Magrath Road, Bengaluru
4		Today	Garuda Mall, Magrath Road, Bengaluru
..			...
195	Today	Whitefield Main Road, Bhoruka Tech Park...	
196		Today	Bellahalli, Bengaluru
197	Today	Whitefield Main Road, Bhoruka Tech Park...	
198		Today	Bellahalli, Bengaluru
199	Today	Whitefield Main Road, Bhoruka Tech Park...	

[200 rows x 4 columns]

```
[49]: import requests
from bs4 import BeautifulSoup
import pandas as pd
from IPython.display import FileLink

def get_page_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Initialize lists to store data
    car_names = []
    car_prices = []
    car_kms = []
    car_locations = []

    # Find all car listings using the structure provided
    cars = soup.find_all('div', class_='_2YB7p')

    print(f"Found {len(cars)} cars on the page.")

    # Loop through each car listing and extract details
    for car in cars:
        name = car.find('h3', class='_11dVb').text if car.find('h3', class='_11dVb') else 'N/A'
        price = car.find('div', class='_2KyOK').text if car.find('div', class='_2KyOK') else 'N/A'

        # Extract kilometers driven from `ul` tag with class `_3J2G-`
        km_tag = car.find('ul', class='_3J2G-')
        km_driven = km_tag.text.strip() if km_tag else 'N/A'
```

```

        # Extract location information from `p` with class `_3dGMY`
        location_tag = car.find('p', class='_3dGMY')
        location_text = ' '.join(span.text for span in location_tag.
↪find_all('span')) if location_tag else 'N/A'

        car_names.append(name)
        car_prices.append(price)
        car_kms.append(km_driven)
        car_locations.append(location_text)

    return car_names, car_prices, car_kms, car_locations

base_url = 'https://www.cars24.com/buy-used-car?
↪sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
↪1724220956&storeCityId=1692&page='

all_car_names = []
all_car_prices = []
all_car_kms = []
all_car_locations = []

# Number of pages to scrape
num_pages = 10

for page in range(1, num_pages + 1):
    url = base_url + str(page)
    print(f"Scraping page {page} with URL: {url}")
    car_names, car_prices, car_kms, car_locations = get_page_data(url)
    all_car_names.extend(car_names)
    all_car_prices.extend(car_prices)
    all_car_kms.extend(car_kms)
    all_car_locations.extend(car_locations)

# Create a DataFrame from the extracted data
df = pd.DataFrame({
    'Car Name': all_car_names,
    'Price': all_car_prices,
    'Kilometers Driven': all_car_kms,
    'Location': all_car_locations
})

# Save DataFrame to CSV
csv_file = 'car_ahmedabad.csv'
df.to_csv(csv_file, index=False)

# Provide a download link

```

```
FileLink(csv_file)
```

Scraping page 1 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=1>

Found 20 cars on the page.

Scraping page 2 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=2>

Found 20 cars on the page.

Scraping page 3 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=3>

Found 20 cars on the page.

Scraping page 4 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=4>

Found 20 cars on the page.

Scraping page 5 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=5>

Found 20 cars on the page.

Scraping page 6 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=6>

Found 20 cars on the page.

Scraping page 7 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=7>

Found 20 cars on the page.

Scraping page 8 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=8>

Found 20 cars on the page.

Scraping page 9 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=9>

Found 20 cars on the page.

Scraping page 10 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=1692&page=10>

Found 20 cars on the page.

[49]: C:\Users\KONGEGOWTAM\car\_ahmedabad.csv

[48]: df

[48]:

	Car Name	Price \
0	2017 Hyundai Elite i20 ASTA 1.2 (O)	5.68 Lakh 6.99 Lakh(1.31L off)
1	2015 Hyundai Elite i20 MAGNA 1.2	4.87 Lakh
2	2016 Maruti Ciaz ZDI SHVS	5.12 Lakh
3	2023 Maruti Alto LXI OPT CNG	4.97 Lakh
4	2023 Tata Tiago XE PETROL	5.16 Lakh 5.57 Lakh(41k off)
..	...	...
195	2022 Hyundai Creta SX 1.5 PETROL	12.77 Lakh
196	2022 Renault TRIBER RXL MT	5.63 Lakh
197	2015 Honda City 1.5L I-VTEC V MT	5.73 Lakh 6.92 Lakh(1.19L off)
198	2020 Maruti Dzire VXI	5.70 Lakh 5.96 Lakh(26.28k off)

199            2023 Honda Amaze 1.2L I-VTEC S            7.45 Lakh 7.70 Lakh(25k off)

	Kilometers Driven		Location
0	83,903 kmPetrolManual	Today	Naroda, Ahmedabad
1	63,192 kmPetrolManual	Today	Naroda, Ahmedabad
2	79,630 kmDieselManual	Today	Naroda, Ahmedabad
3	18,726 kmCNGManual	Today	Karnavati Club, Ahmedabad
4	5,985 kmPetrolManual	Today	Naroda, Ahmedabad
..	...		...
195	43,583 kmPetrolManual	Today	Karnavati Club, Ahmedabad
196	26,543 kmPetrolManual	Today	Karnavati Club, Ahmedabad
197	49,727 kmPetrolManual	Today	Naroda, Ahmedabad
198	81,469 kmPetrolManual	Today	Naroda, Ahmedabad
199	11,259 kmPetrolManual	Today	Naroda, Ahmedabad

[200 rows x 4 columns]

```
[56]: import requests
from bs4 import BeautifulSoup
import pandas as pd
from IPython.display import FileLink

def get_page_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Initialize lists to store data
    car_names = []
    car_prices = []
    car_kms = []
    car_locations = []

    # Find all car listings using the structure provided
    cars = soup.find_all('div', class_='_2YB7p')

    print(f"Found {len(cars)} cars on the page.")

    # Loop through each car listing and extract details
    for car in cars:
        name = car.find('h3', class_='_11dVb').text if car.find('h3',
↪class_='_11dVb') else 'N/A'
        price = car.find('div', class_='_2KyOK').text if car.find('div',
↪class_='_2KyOK') else 'N/A'

        # Extract kilometers driven from `ul` tag with class `_3J2G-`
        km_tag = car.find('ul', class_='_3J2G-')
        km_driven = km_tag.text.strip() if km_tag else 'N/A'
```

```

        # Extract location information from `p` with class `_3dGMY`
        location_tag = car.find('p', class='_3dGMY')
        location_text = ' '.join(span.text for span in location_tag.
↪find_all('span')) if location_tag else 'N/A'

        car_names.append(name)
        car_prices.append(price)
        car_kms.append(km_driven)
        car_locations.append(location_text)

    return car_names, car_prices, car_kms, car_locations

base_url = 'https://www.cars24.com/buy-used-car?
↪sort=bestmatch&serveWarrantyCount=true&gaId=283731005.
↪1724220956&storeCityId=5732&page='

all_car_names = []
all_car_prices = []
all_car_kms = []
all_car_locations = []

# Number of pages to scrape
num_pages = 10

for page in range(1, num_pages + 1):
    url = base_url + str(page)
    print(f"Scraping page {page} with URL: {url}")
    car_names, car_prices, car_kms, car_locations = get_page_data(url)
    all_car_names.extend(car_names)
    all_car_prices.extend(car_prices)
    all_car_kms.extend(car_kms)
    all_car_locations.extend(car_locations)

# Create a DataFrame from the extracted data
df = pd.DataFrame({
    'Car Name': all_car_names,
    'Price': all_car_prices,
    'Kilometers Driven': all_car_kms,
    'Location': all_car_locations
})

# Save DataFrame to CSV
csv_file = 'car_Chennai.csv'
df.to_csv(csv_file, index=False)

# Provide a download link

```

```
FileLink(csv_file)
```

Scraping page 1 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=1>

Found 20 cars on the page.

Scraping page 2 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=2>

Found 20 cars on the page.

Scraping page 3 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=3>

Found 20 cars on the page.

Scraping page 4 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=4>

Found 20 cars on the page.

Scraping page 5 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=5>

Found 20 cars on the page.

Scraping page 6 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=6>

Found 20 cars on the page.

Scraping page 7 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=7>

Found 20 cars on the page.

Scraping page 8 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=8>

Found 20 cars on the page.

Scraping page 9 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=9>

Found 20 cars on the page.

Scraping page 10 with URL: <https://www.cars24.com/buy-used-car?sort=bestmatch&serveWarrantyCount=true&gaId=283731005.1724220956&storeCityId=5732&page=10>

Found 20 cars on the page.

[56]: C:\Users\KONGEGOWTAM\car\_Chennai.csv

[55]: df

[55]:

	Car Name \
0	2016 Tata Tiago XZ PETROL
1	2015 Hyundai i10 SPORTZ 1.1
2	2015 Maruti Swift VXI
3	2017 Datsun Redi Go T (0)
4	2021 Tata Tiago XE PETROL
..	...
195	2020 Hyundai AURA SX 1.2
196	2018 Maruti IGNIS DELTA 1.2
197	2021 Toyota URBAN CRUISER HIGH GRADE MT
198	2019 Renault Kwid RXT 1.0 (0)



199 2018 Volkswagen Vento HIGHLINE PETROL AT

	Price	Kilometers Driven \
0	4.05 Lakh	77,078 kmPetrolManual
1	3.90 Lakh	36,748 kmPetrolManual
2	4.47 Lakh	79,587 kmPetrolManual
3	2.66 Lakh 3.13 Lakh(47.26k off)	29,913 kmPetrolManual
4	4.57 Lakh	43,193 kmPetrolManual
..	...	...
195	6.60 Lakh	41,631 kmPetrolManual
196	4.60 Lakh	81,794 kmPetrolManual
197	8.45 Lakh 8.60 Lakh(15k off)	16,303 kmPetrolManual
198	3.93 Lakh 5.10 Lakh(1.17L off)	50,692 kmPetrolManual
199	8.00 Lakh	52,705 kmPetrolAutomatic

	Location
0	Today Navalur, Chennai
1	Today Thiruverkadu, Chennai\n
2	Today Navalur, Chennai
3	Today Thiruverkadu, Chennai\n
4	Today Navalur, Chennai
..	...
195	Today Navalur, Chennai
196	Today Navalur, Chennai
197	Today Thiruverkadu, Chennai\n
198	Today Thiruverkadu, Chennai\n
199	Today Navalur, Chennai

[200 rows x 4 columns]

[ ]: