Python

Web Scrapping

Outline

- 3 Popular Tools and Libraries used for Web Scraping in Python
- Basic Data Scrapping
- Web Scraping
  - Crawl
  - Parse and Transform
  - Store
- Scraping Images

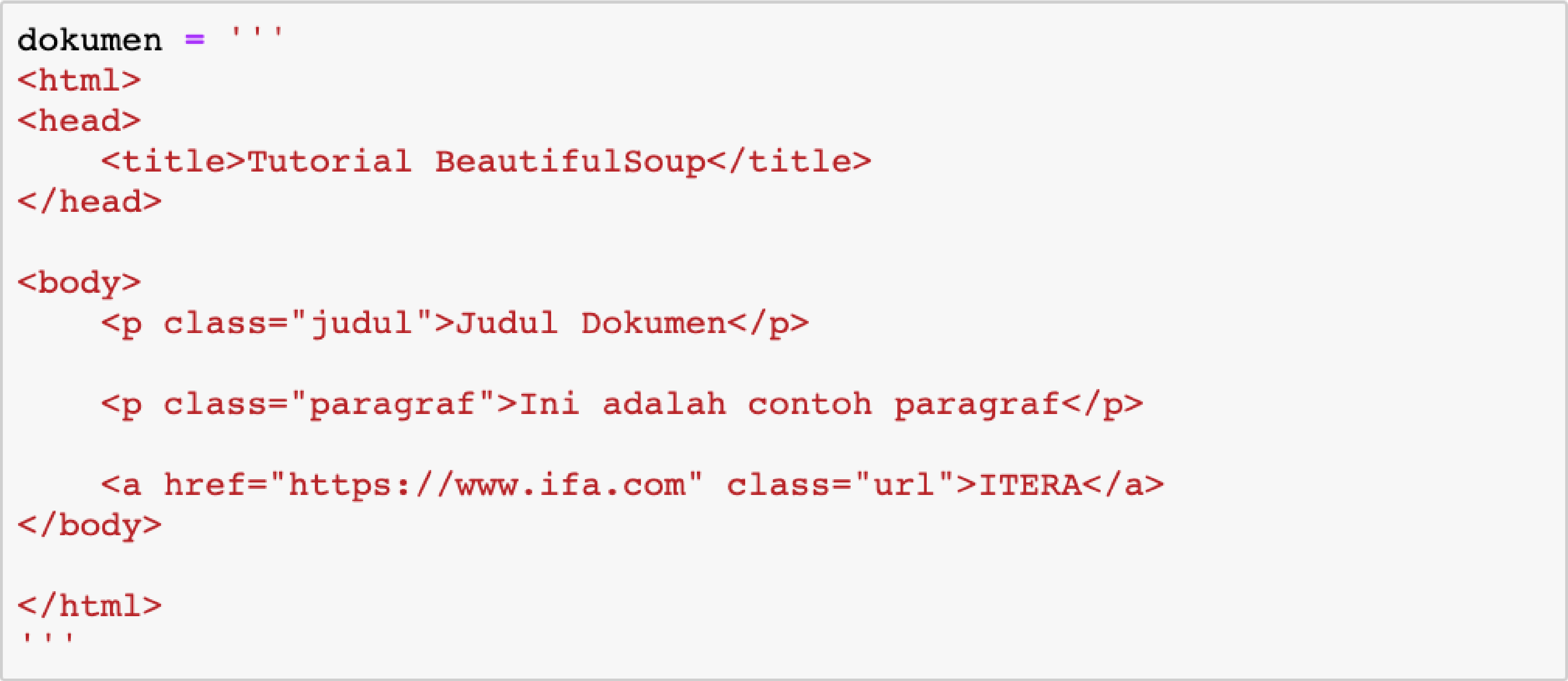
Three populars library:

- **BeautifulSoup**
  - *BeautifulSoup* is an amazing parsing library in Python that enables us to extract data from HTML and XML documents.
- **Scrapy**
  - Scrapy is a Python framework for large scale web scraping.
- **Selenium**
  - Selenium is another popular tool for automating browsers. It's primarily used for testing in the industry but is also very handy for web scraping.

Basic Data Scrapping

Data Scrapping

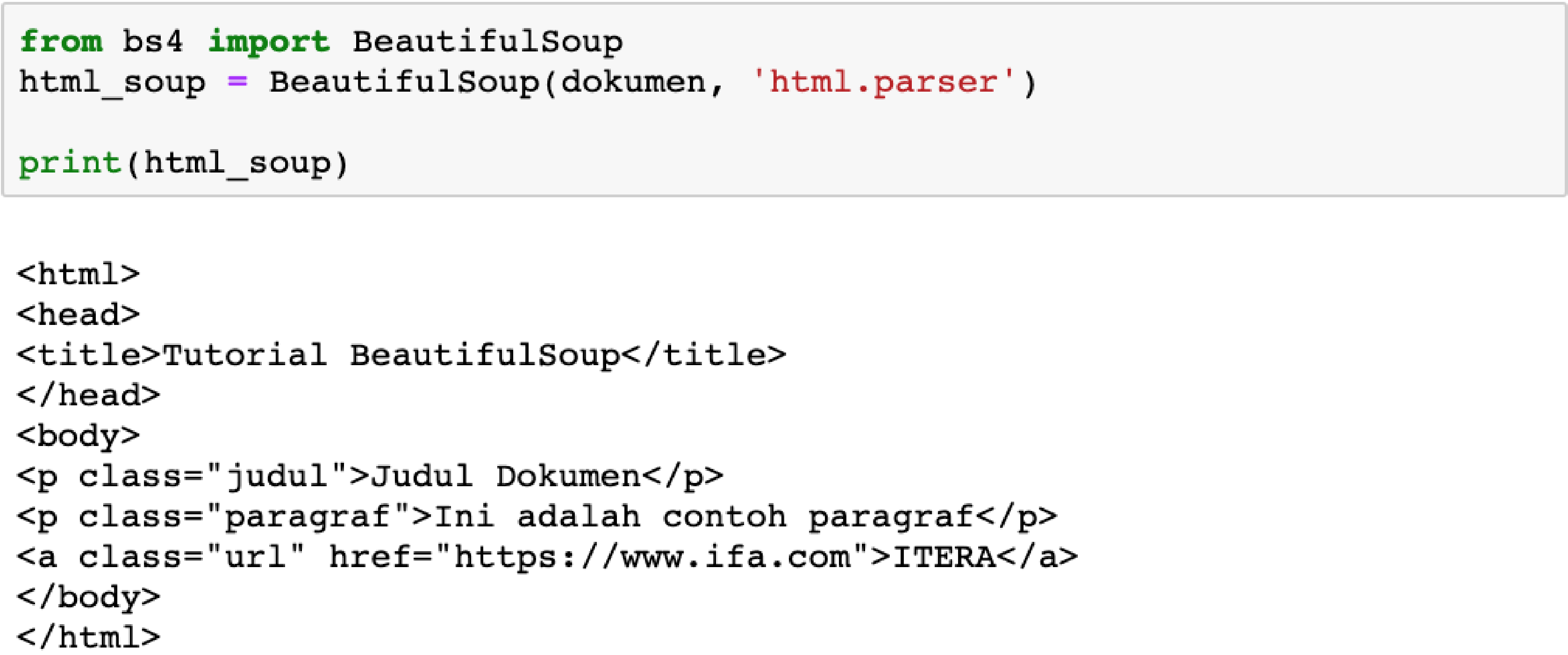
- Create code like this.



```
dokumen = '''
<html>
<head>
    <title>Tutorial BeautifulSoup</title>
</head>

<body>
    <p class="judul">Judul Dokumen</p>

    <p class="paragraf">Ini adalah contoh paragraf</p>

    <a href="https://www.ifa.com" class="url">ITERA</a>
</body>

</html>
'''
```

Print Data using HTML Parser

- To use HTML parser, we need BeautifulSoup library. By default, BeautifulSoup has been availabled in Anaconda (Jupyter Notebook)

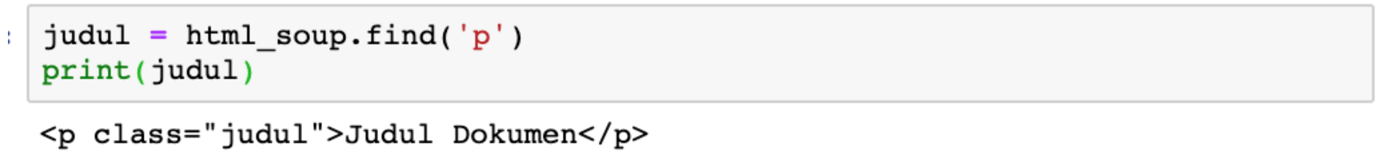


```
from bs4 import BeautifulSoup
html_soup = BeautifulSoup(dokumen, 'html.parser')

print(html_soup)
```

```
<html>
<head>
<title>Tutorial BeautifulSoup</title>
</head>
<body>
<p class="judul">Judul Dokumen</p>
<p class="paragraf">Ini adalah contoh paragraf</p>
<a class="url" href="https://www.ifa.com">ITERA</a>
</body>
</html>
```

Get Data From Web Document

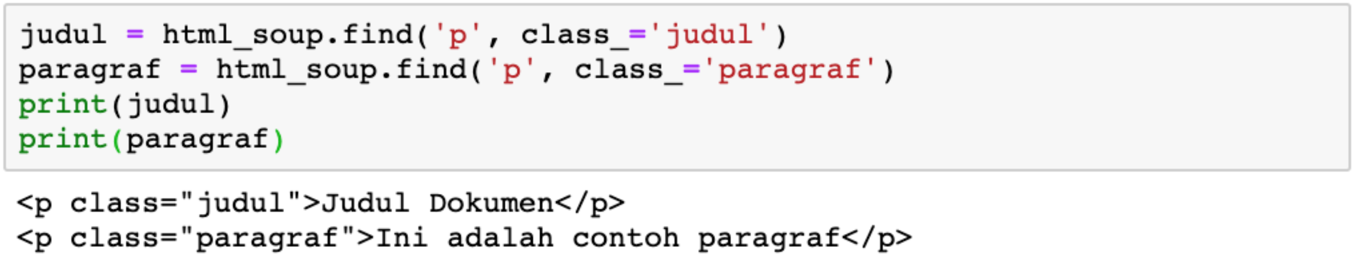
- Get HTML data From Element)



```
judul = html_soup.find('p')
print(judul)
```

```
<p class="judul">Judul Dokumen</p>
```

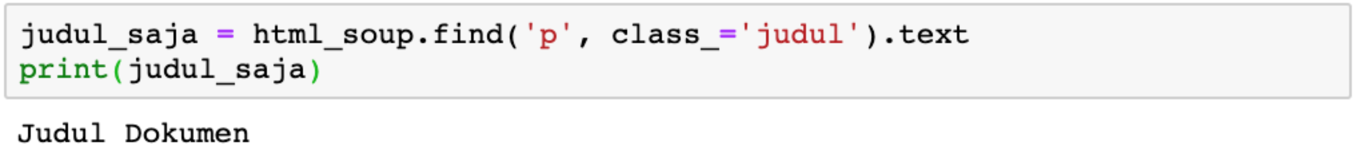
- Get HTML data from element using **class**



```
judul = html_soup.find('p', class_='judul')
paragraf = html_soup.find('p', class_='paragraf')
print(judul)
print(paragraf)
```

```
<p class="judul">Judul Dokumen</p>
<p class="paragraf">Ini adalah contoh paragraf</p>
```

- Get text in class ‘judul’ with element ‘p’

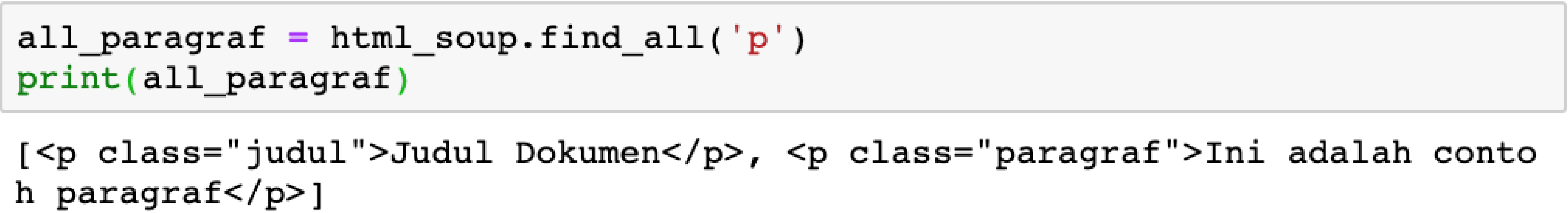


```
judul_saja = html_soup.find('p', class_='judul').text
print(judul_saja)
```

```
Judul Dokumen
```

Find Data or element or class in web site

- The find () function can only extract one output while usually many of the same HTML tags that all of them want to retrieve.
- To retrieve HTML content with the same tag use the find_all () function

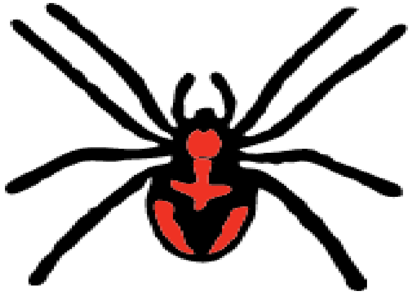


```python
all_paragraf = html_soup.find_all('p')
print(all_paragraf)
```

```
[<p class="judul">Judul Dokumen</p>, <p class="paragraf">Ini adalah conto
h paragraf</p>]
```

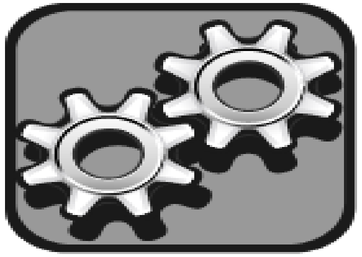
Web Scrapping

Components of Web Scraping



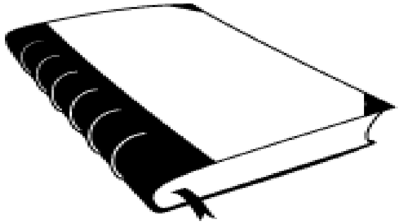
1. **Crawl**

   The first step is always navigate to the target website by making an HTTP Request and download the response you get.



2. **Parse and Transform**

   Once you have received the response, Now its time to parse this downloaded data into a HTML Parser like Beautiful Soup and Extract the Required Data.

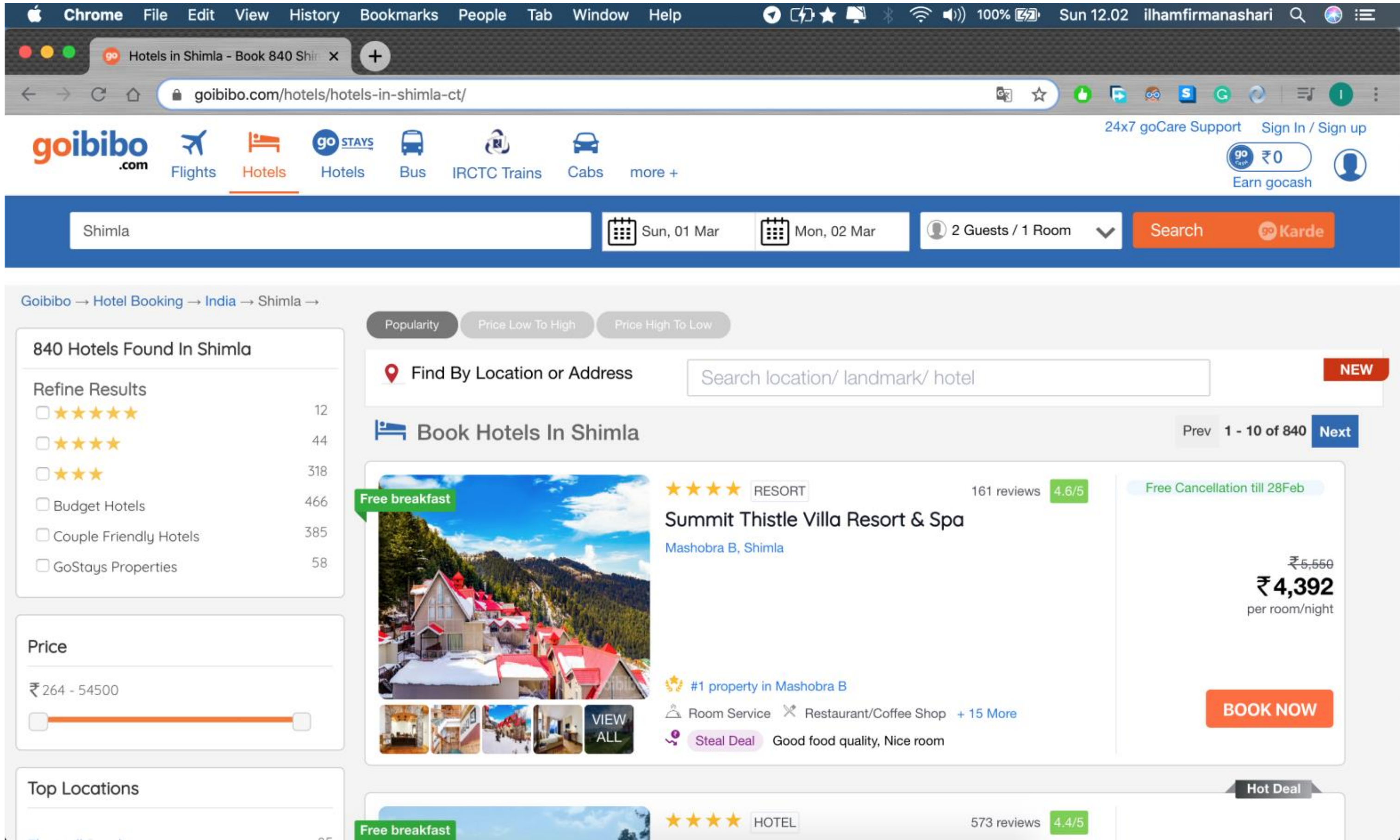


3. **Store**

   Now that you have extracted the required data. You can simply store this as JSON or CSV file or directly into the the DataBase like MongoDB

Get-Started

- Let’s understand these components in detail. We’ll do this by scraping hotel details like the name of the hotel and price per room from the ***goibibo* website**: goibibo.com/hotels/hotels-in-shimla-ct/



Shimla | Sun, 01 Mar | Mon, 02 Mar | 2 Guests / 1 Room

Goibibo → Hotel Booking → India → Shimla →

840 Hotels Found In Shimla

Refine Results

□ ★★★★★	12
□ ★★★★	44
□ ★★★	318
□ Budget Hotels	466
□ Couple Friendly Hotels	385
□ GoStays Properties	58

Price

₹264 - 54500

Top Locations

Find By Location or Address: Search location/ landmark/ hotel

Book Hotels In Shimla

1 - 10 of 840

Free breakfast

★★★★ RESORT 161 reviews 4.6/5

Summit Thistle Villa Resort & Spa

Mashobra B, Shimla

Free Cancellation till 28Feb

~~₹5,550~~
₹4,392
per room/night

#1 property in Mashobra B

Room Service · Restaurant/Coffee Shop · + 15 More

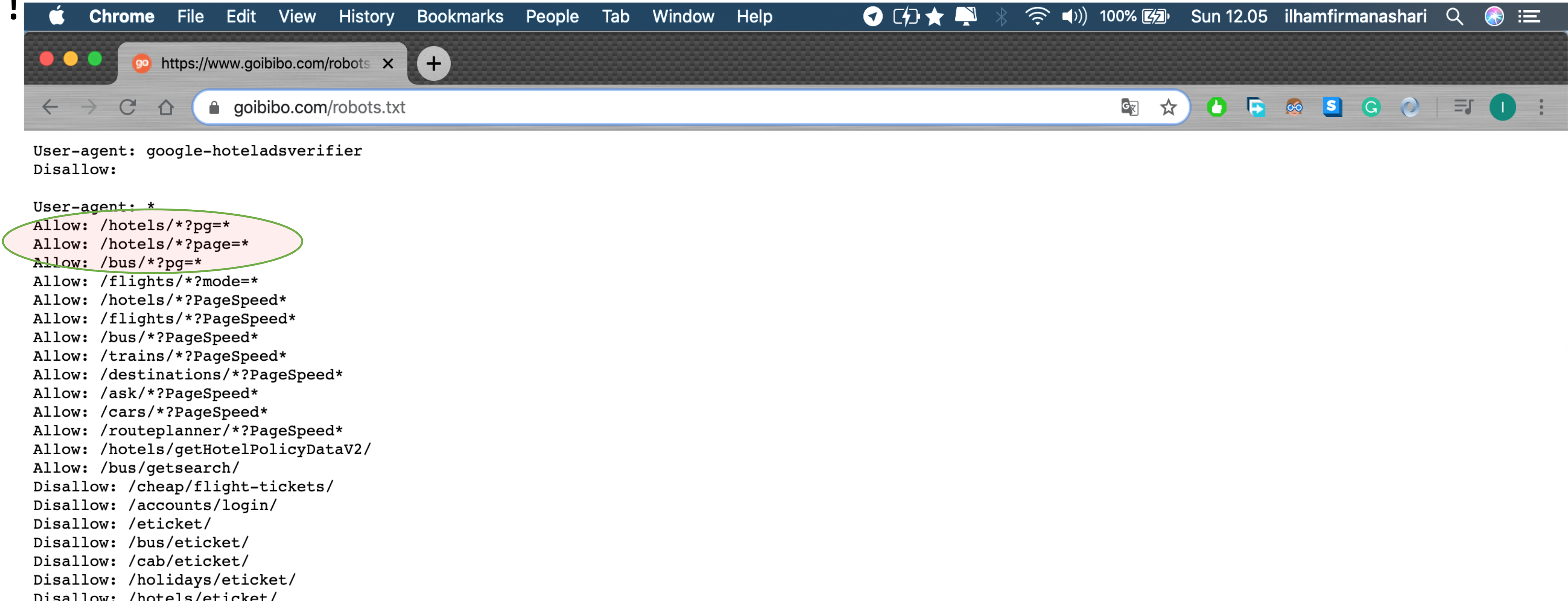
Steal Deal · Good food quality, Nice room

Free breakfast

★★★★ HOTEL 573 reviews 4.4/5

Analysis before scrapping using (**robots.txt)**

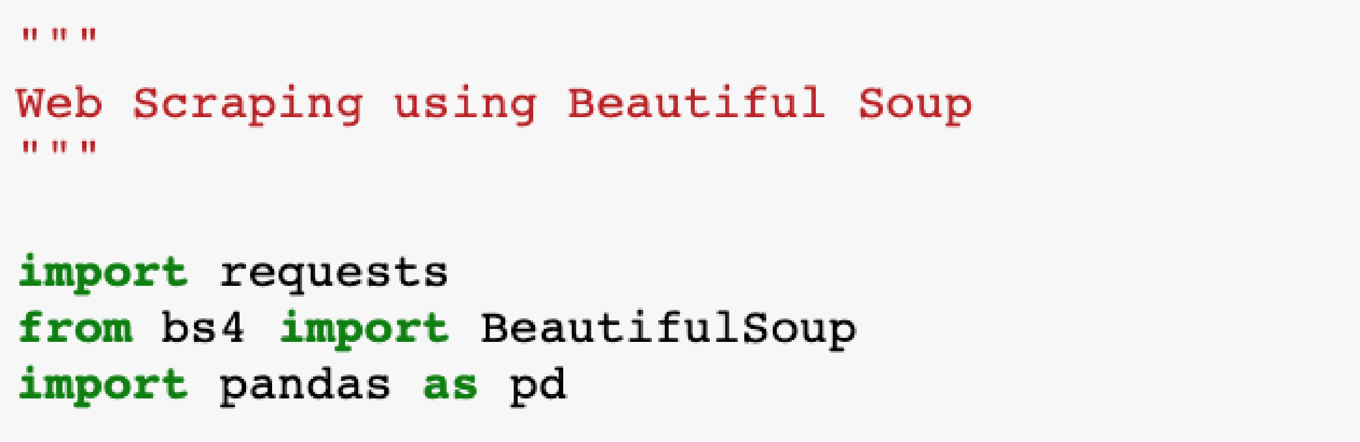
- So, looks like we are allowed to scrape the data from our targeted URL. We are good to go and write the script of our web robot. Let’s begin!



```
User-agent: google-hoteladsverifier
Disallow:

User-agent: *
Allow: /hotels/*?pg=*
Allow: /hotels/*?page=*
Allow: /bus/*?pg=*
Allow: /flights/*?mode=*
Allow: /hotels/*?PageSpeed*
Allow: /flights/*?PageSpeed*
Allow: /bus/*?PageSpeed*
Allow: /trains/*?PageSpeed*
Allow: /destinations/*?PageSpeed*
Allow: /ask/*?PageSpeed*
Allow: /cars/*?PageSpeed*
Allow: /routeplanner/*?PageSpeed*
Allow: /hotels/getHotelPolicyDataV2/
Allow: /bus/getsearch/
Disallow: /cheap/flight-tickets/
Disallow: /accounts/login/
Disallow: /eticket/
Disallow: /bus/eticket/
Disallow: /cab/eticket/
Disallow: /holidays/eticket/
Disallow: /hotels/eticket/
```

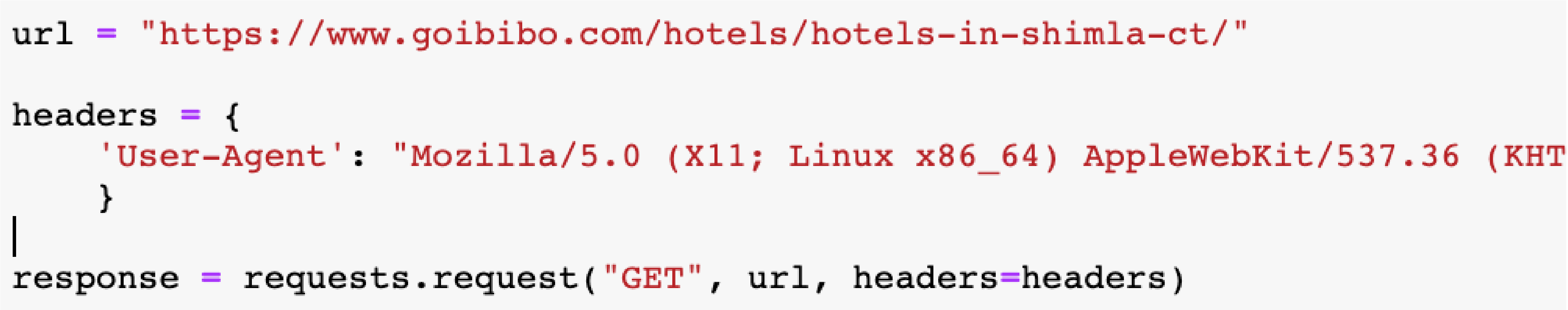
Step 1 : Start to Scrapping
Define libraries

- To scrapping the data from website, we need 3 libraries



```
"""
Web Scraping using Beautiful Soup
"""

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

Step 1 : Start to Scrapping
URL Web that will be scrapped

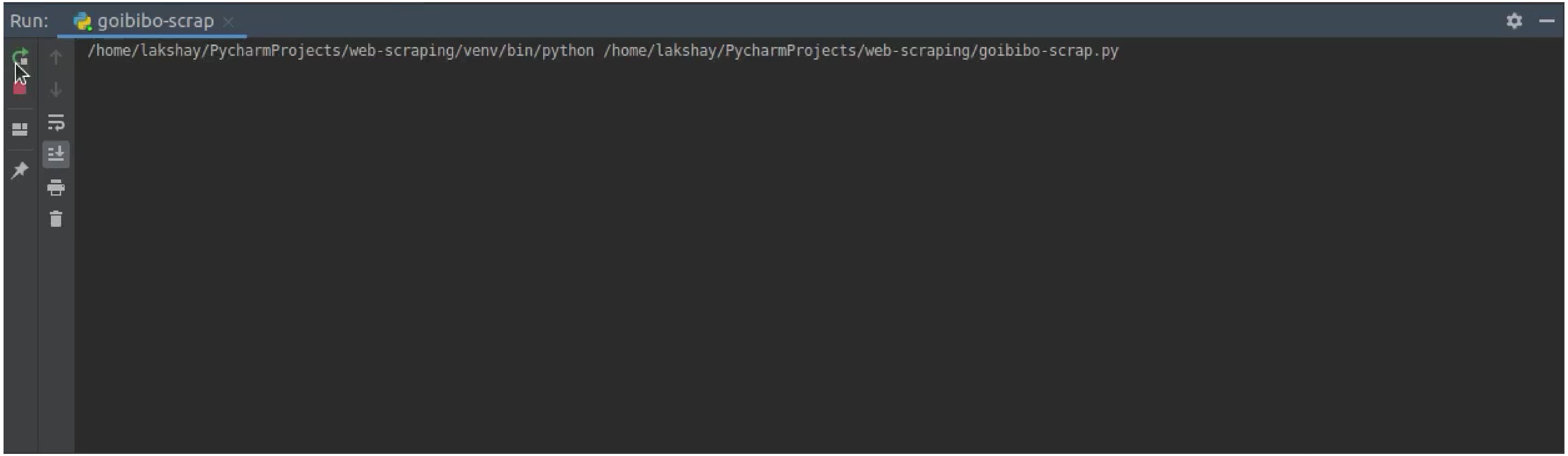


```
url = "https://www.goibibo.com/hotels/hotels-in-shimla-ct/"

headers = {
    'User-Agent': "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHT
    }

response = requests.request("GET", url, headers=headers)
```

Note: to know the headers of the website, inspect element.
optional

Step 2 : Parsing and Transform
Parsing html (web-document)

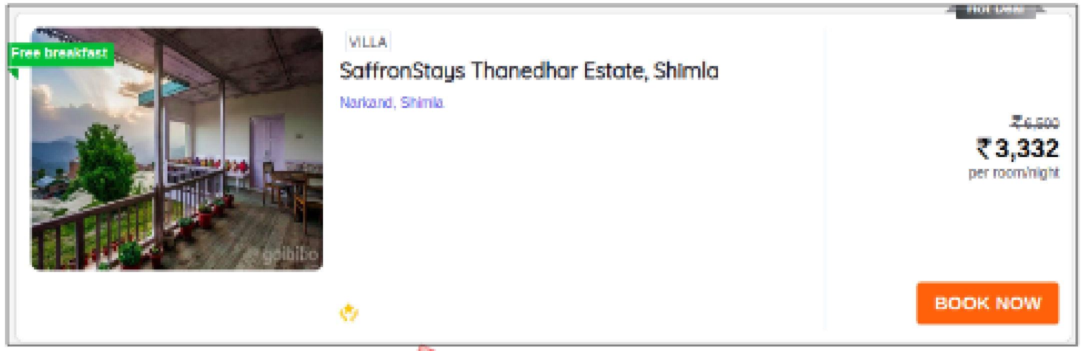
```
data = BeautifulSoup(response.text, 'html.parser')
print(data)
```

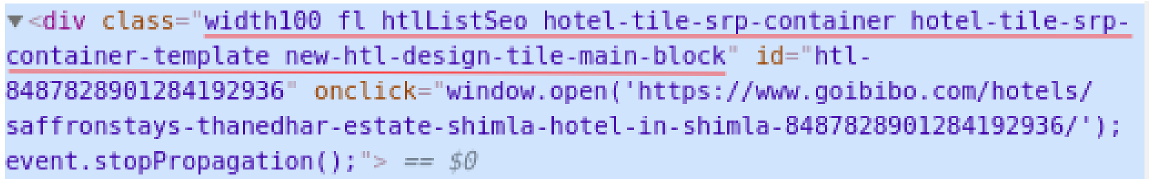


```
/home/lakshay/PycharmProjects/web-scraping/venv/bin/python /home/lakshay/PycharmProjects/web-scraping/goibibo-scrap.py
```

Step 2: Parsing and Transform Inspect element

- The next step is to parse this data into an HTML Parser and for that, we will use the *BeautifulSoup* library.
- Next, we will select the card and click on the 'Inspect Element' option to get the source code of that particular card. You will get something like this:

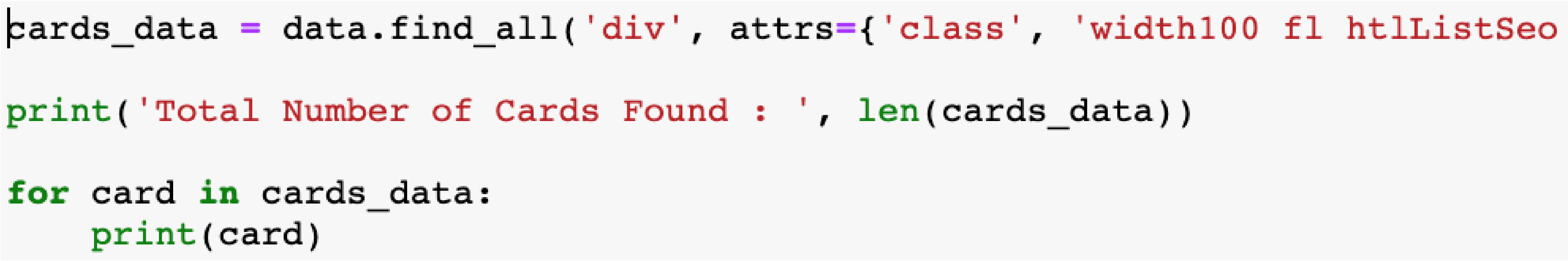




```
▼<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-
container-template new-htl-design-tile-main-block" id="htl-
8487828901284192936" onclick="window.open('https://www.goibibo.com/hotels/
saffronstays-thanedhar-estate-shimla-hotel-in-shimla-8487828901284192936/');
event.stopPropagation();"> == $0
```

Step 2: Parsing and Transform
Get root element of the data

- The class name of all the cards would be the same and we can get a list of those cards by just passing the tag name and attributes like the *<class>* tag with its name like I've shown below:

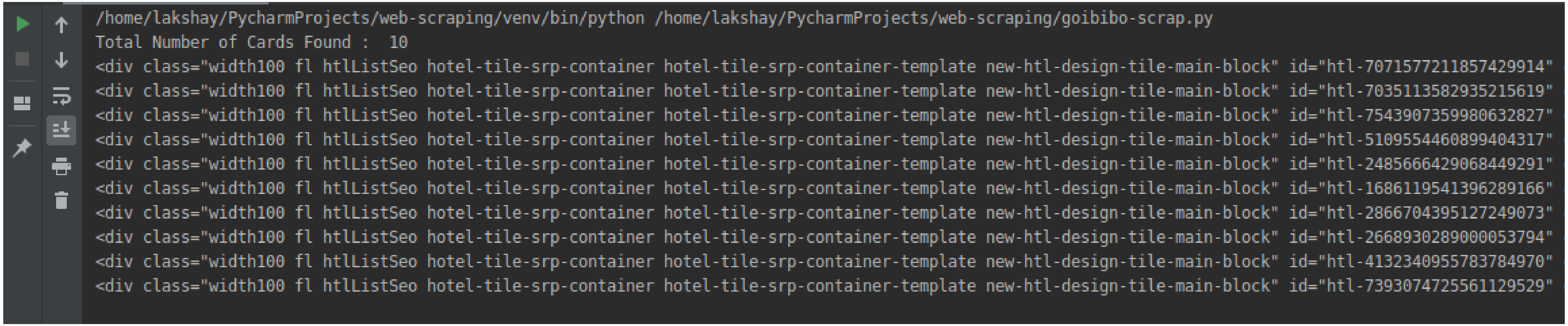


```
cards_data = data.find_all('div', attrs={'class', 'width100 fl htlListSeo

print('Total Number of Cards Found : ', len(cards_data))

for card in cards_data:
    print(card)
```

- **Note:** for more clearly, do inspect element in your web browser to get full class name

Step 2: Parsing and Transform

Result

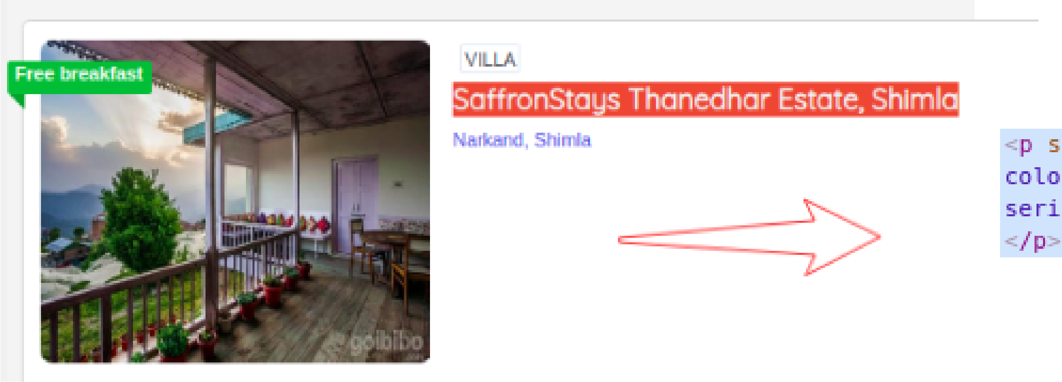


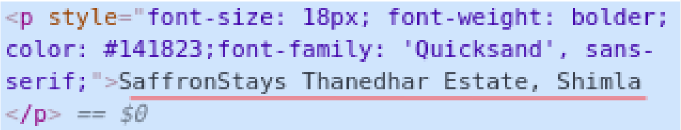
```
/home/lakshay/PycharmProjects/web-scraping/venv/bin/python /home/lakshay/PycharmProjects/web-scraping/goibibo-scrap.py
Total Number of Cards Found :  10
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-7071577211857429914"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-7035113582935215619"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-7543907359980632827"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-5109554460899404317"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-2485666429068449291"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-1686119541396289166"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-2866704395127249073"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-2668930289000053794"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-4132340955783784970"
<div class="width100 fl htlListSeo hotel-tile-srp-container hotel-tile-srp-container-template new-htl-design-tile-main-block" id="htl-7393074725561129529"
```

Step 2: Parsing and Transform

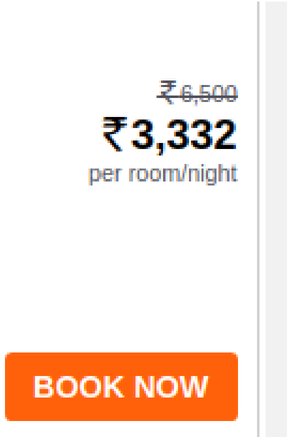
Get Hotel name and Room Price

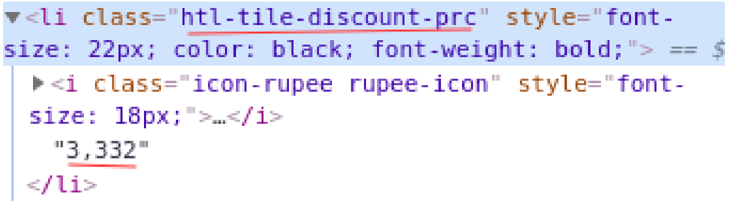
- Select only the Hotel Name, perform the Inspect Element step, and do the same with the Room Price:





```
<p style="font-size: 18px; font-weight: bolder;
color: #141823;font-family: 'Quicksand', sans-
serif;">SaffronStays Thanedhar Estate, Shimla
</p> == $0
```



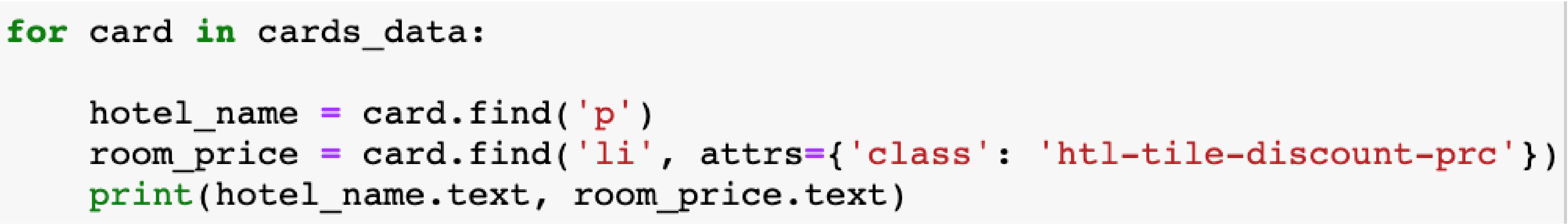


```
▼<li class="htl-tile-discount-prc" style="font-
size: 22px; color: black; font-weight: bold;"> == $
  ▶<i class="icon-rupee rupee-icon" style="font-
  size: 18px;">…</i>
    "3,332"
  </li>
```

Step 2: Parsing and Transform

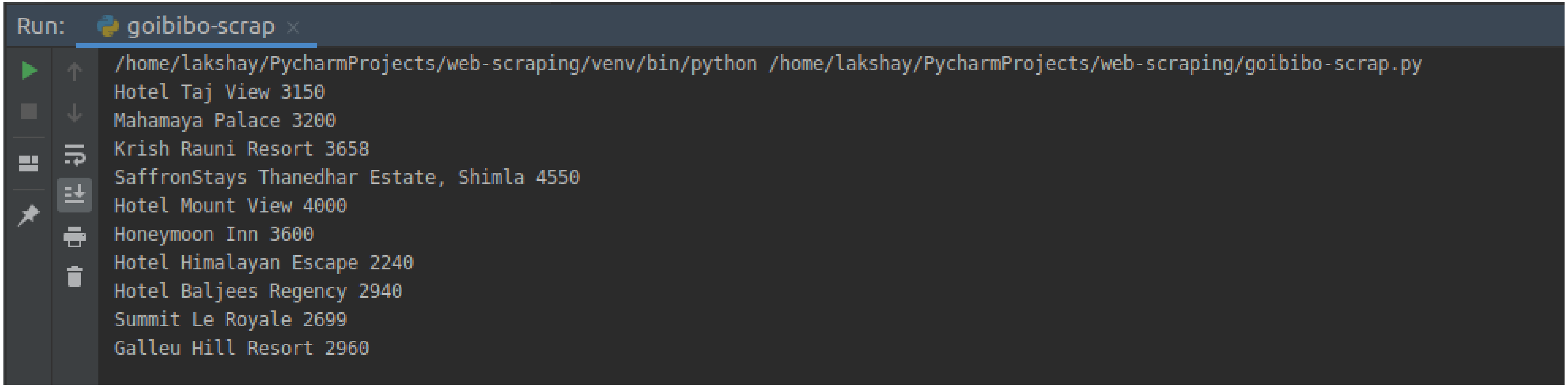
Get element in web for the data

- Now, for each card, we have to find the Hotel Name which can be extracted from the *<p>* tag .
- **This is because there is only one *<p>* tag for each card and Room Price by <li> tag along with the <class> tag and class name:**



```
for card in cards_data:

    hotel_name = card.find('p')
    room_price = card.find('li', attrs={'class': 'htl-tile-discount-prc'})
    print(hotel_name.text, room_price.text)
```

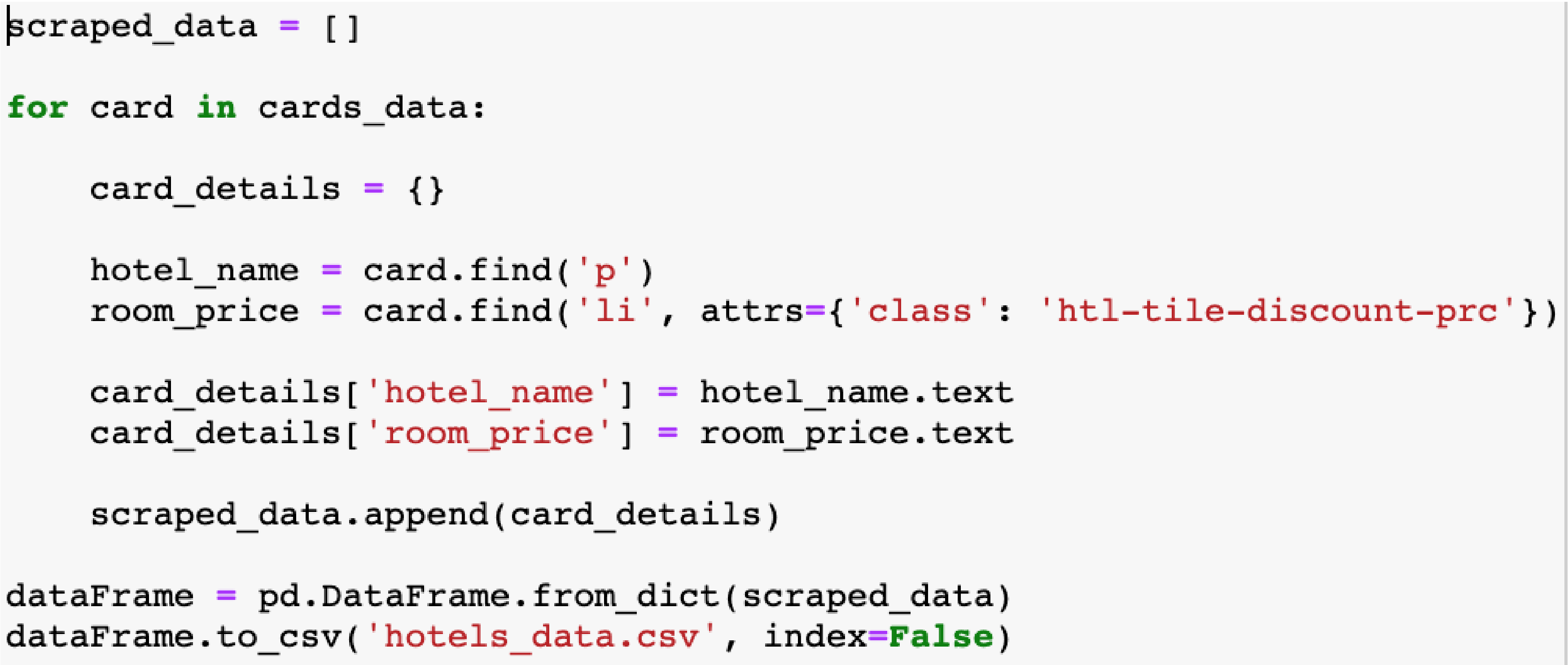
Step 2: Parsing and Transform
result



```
/home/lakshay/PycharmProjects/web-scraping/venv/bin/python /home/lakshay/PycharmProjects/web-scraping/goibibo-scrap.py
Hotel Taj View 3150
Mahamaya Palace 3200
Krish Rauni Resort 3658
SaffronStays Thanedhar Estate, Shimla 4550
Hotel Mount View 4000
Honeymoon Inn 3600
Hotel Himalayan Escape 2240
Hotel Baljees Regency 2940
Summit Le Royale 2699
Galleu Hill Resort 2960
```

Step 3 : Store the Data

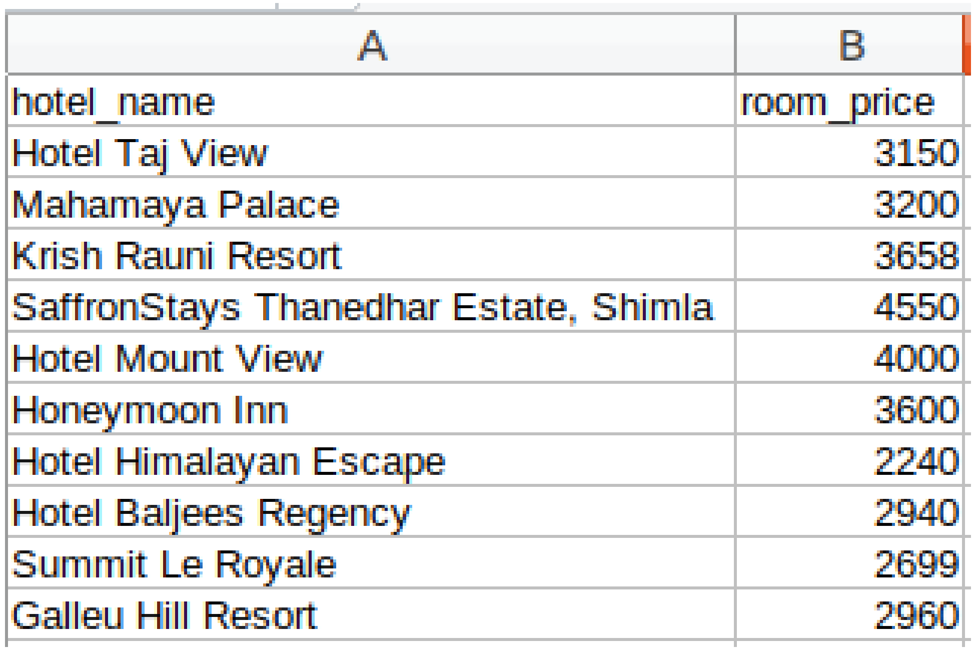
- Next, let’s go ahead and transform this list to a Pandas dataframe as it allows us to convert the dataframe into CSV or JSON files:



```
scraped_data = []

for card in cards_data:

    card_details = {}

    hotel_name = card.find('p')
    room_price = card.find('li', attrs={'class': 'htl-tile-discount-prc'})

    card_details['hotel_name'] = hotel_name.text
    card_details['room_price'] = room_price.text

    scraped_data.append(card_details)

dataFrame = pd.DataFrame.from_dict(scraped_data)
dataFrame.to_csv('hotels_data.csv', index=False)
```

Result

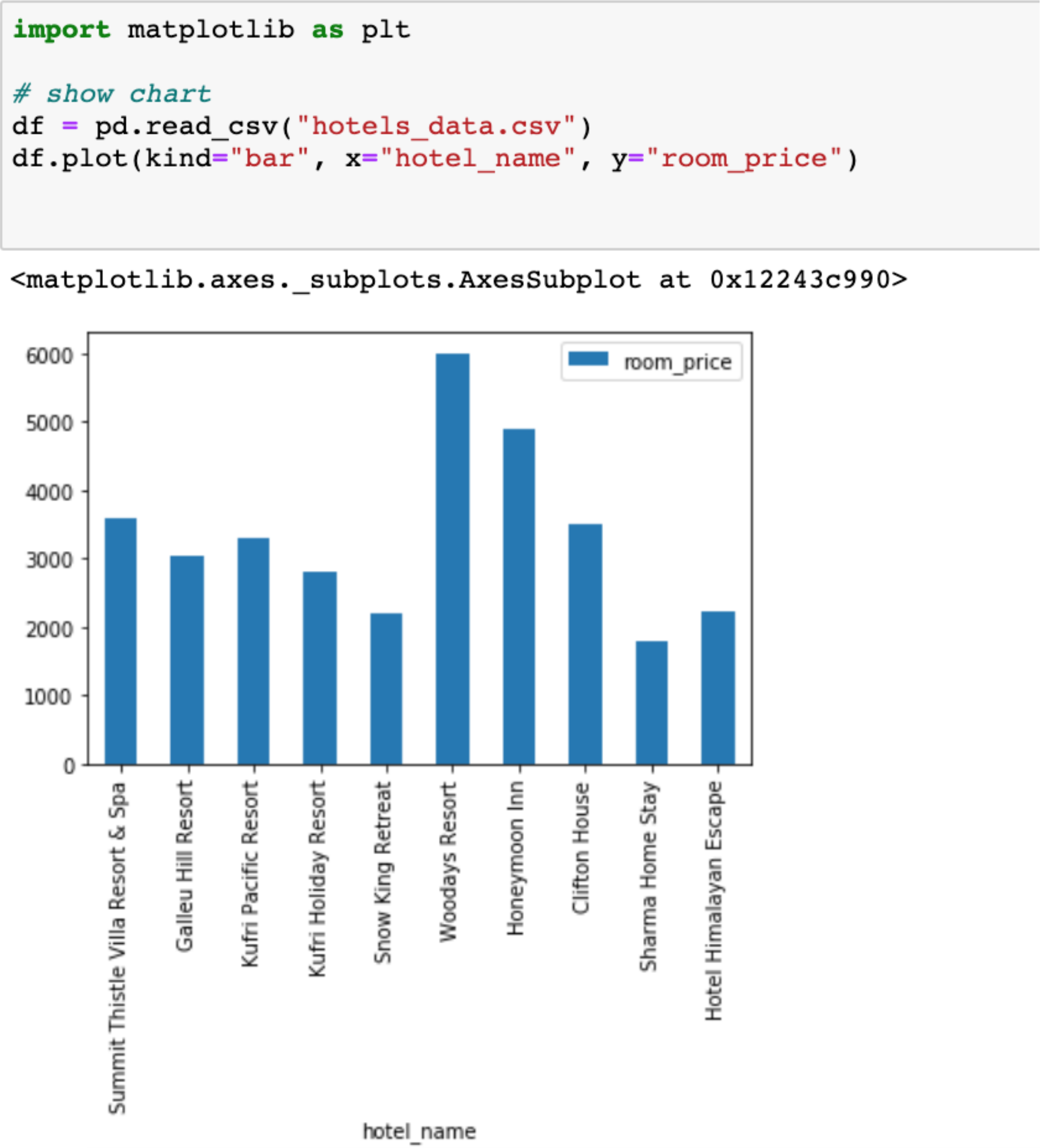
You can use this file to data analysis(create images plot, etc)



A	B
hotel_name	room_price
Hotel Taj View	3150
Mahamaya Palace	3200
Krish Rauni Resort	3658
SaffronStays Thanedhar Estate, Shimla	4550
Hotel Mount View	4000
Honeymoon Inn	3600
Hotel Himalayan Escape	2240
Hotel Baljees Regency	2940
Summit Le Royale	2699
Galleu Hill Resort	2960

Data Visualization

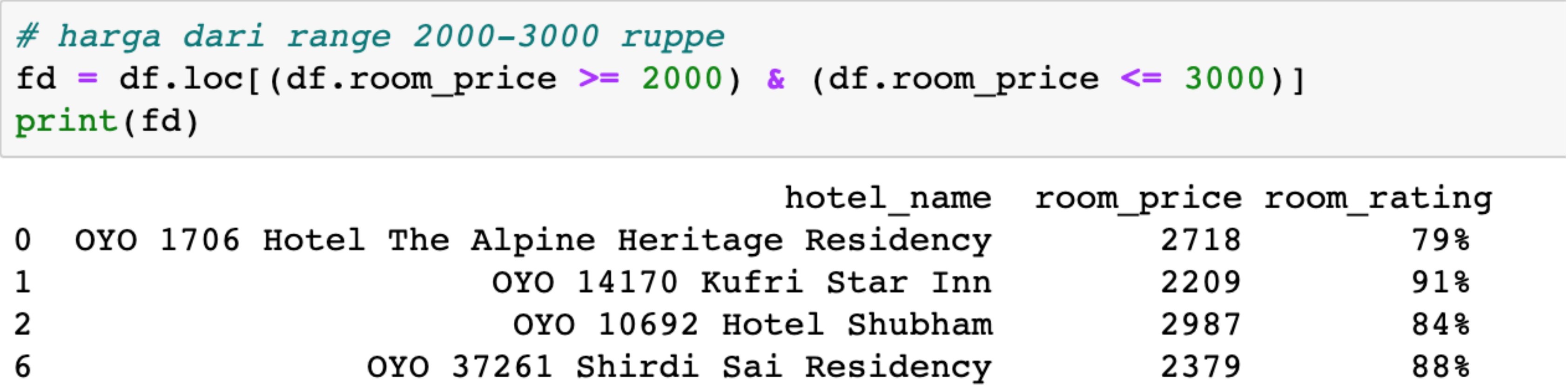
- Show the room price on each hotel



```
import matplotlib as plt

# show chart
df = pd.read_csv("hotels_data.csv")
df.plot(kind="bar", x="hotel_name", y="room_price")
```

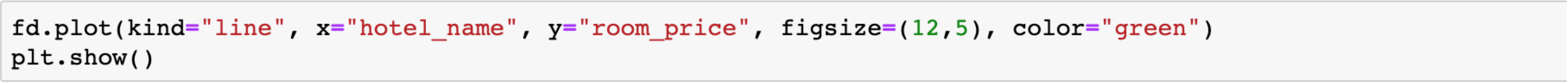
```
<matplotlib.axes._subplots.AxesSubplot at 0x12243c990>
```

Data Visualization

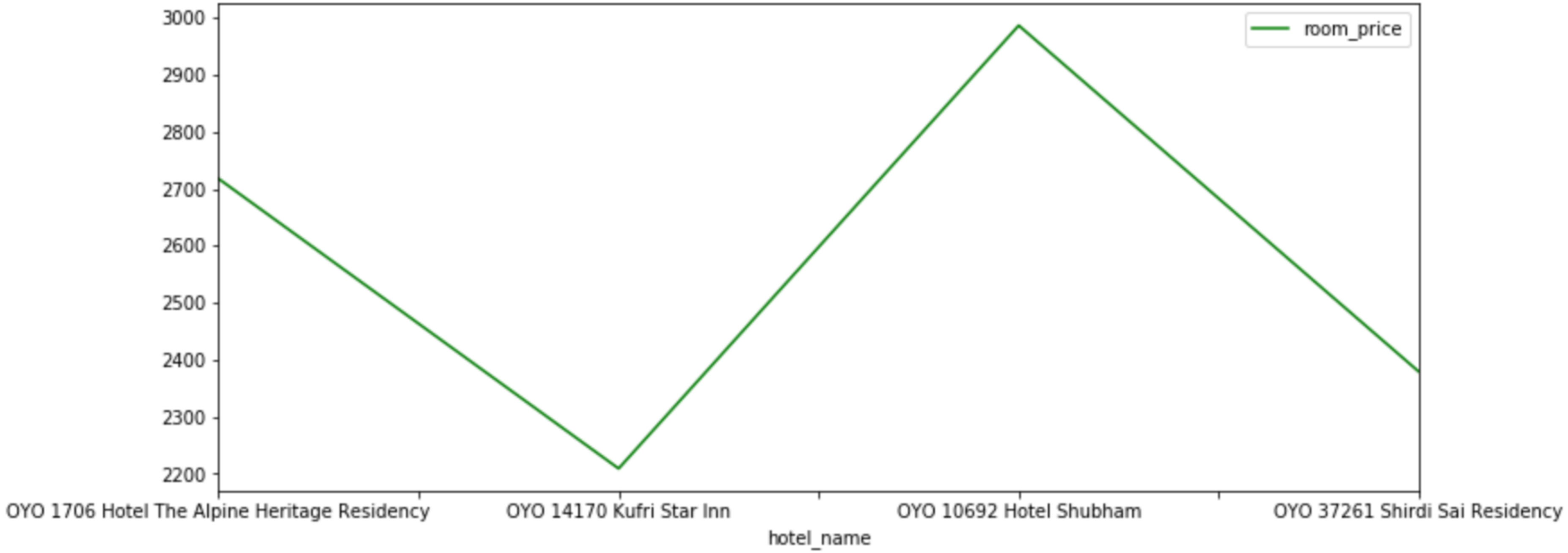


```
# harga dari range 2000-3000 ruppe
fd = df.loc[(df.room_price >= 2000) & (df.room_price <= 3000)]
print(fd)
```

```
                                   hotel_name  room_price room_rating
0  OYO 1706 Hotel The Alpine Heritage Residency        2718         79%
1                         OYO 14170 Kufri Star Inn        2209         91%
2                          OYO 10692 Hotel Shubham        2987         84%
6                    OYO 37261 Shirdi Sai Residency        2379         88%
```

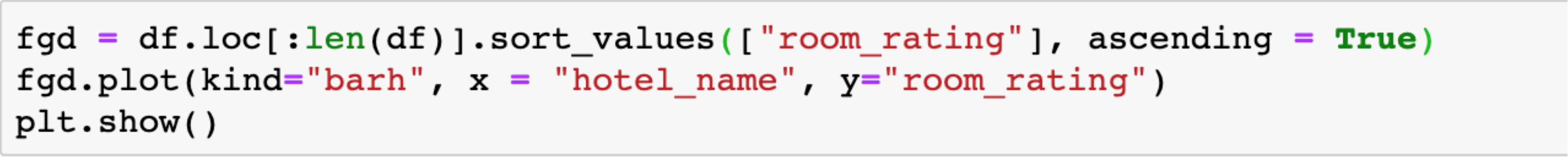


```
fd.plot(kind="line", x="hotel_name", y="room_price", figsize=(12,5), color="green")
plt.show()
```

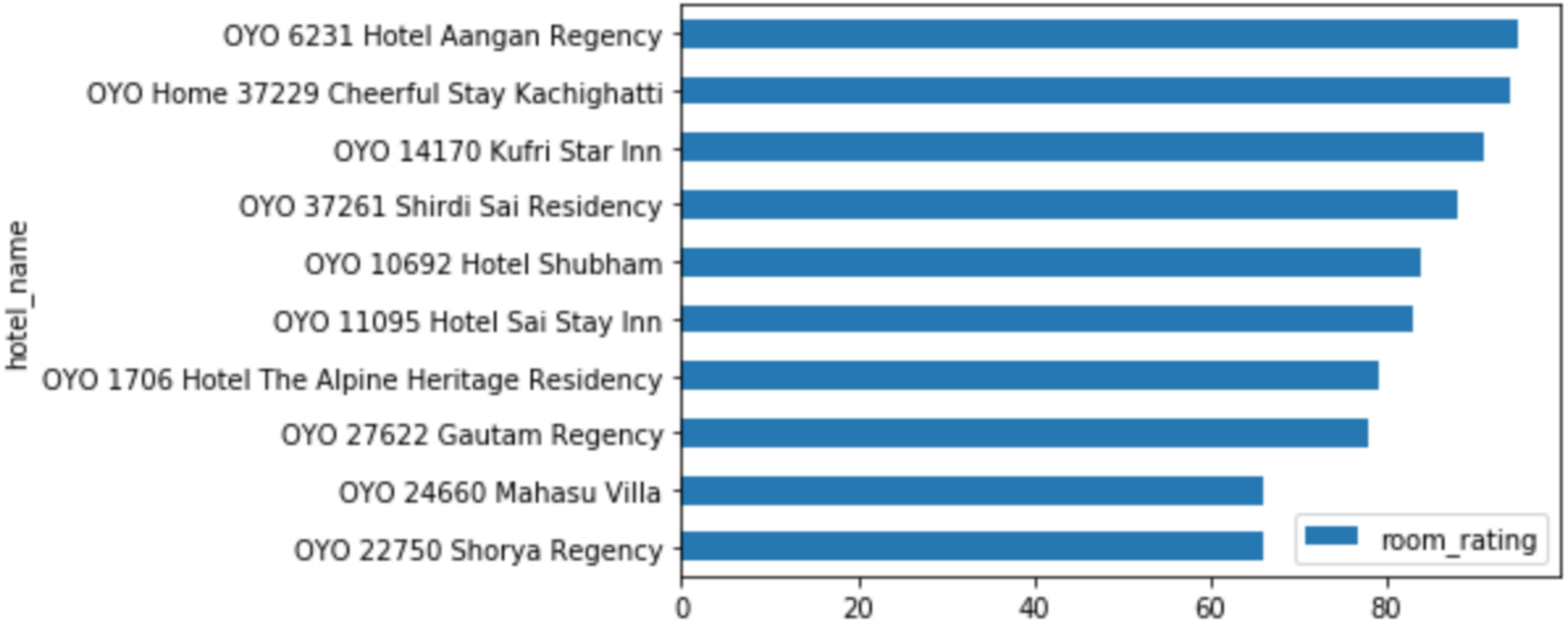


Data Visualization

- Show the highest hotel rating



```
fgd = df.loc[:len(df)].sort_values(["room_rating"], ascending = True)
fgd.plot(kind="barh", x = "hotel_name", y="room_rating")
plt.show()
```

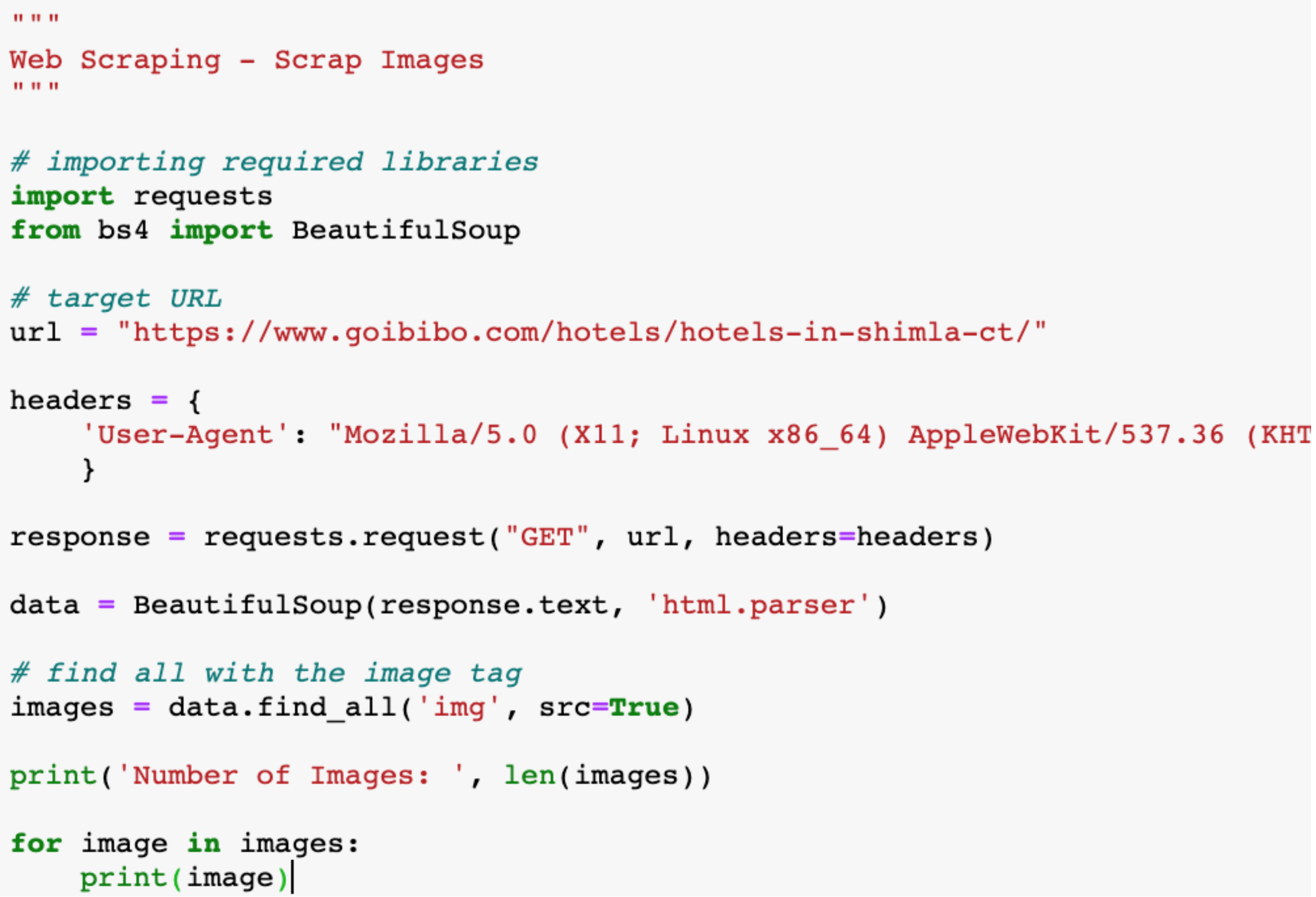


Scrapping Images

Scrape Images in Python

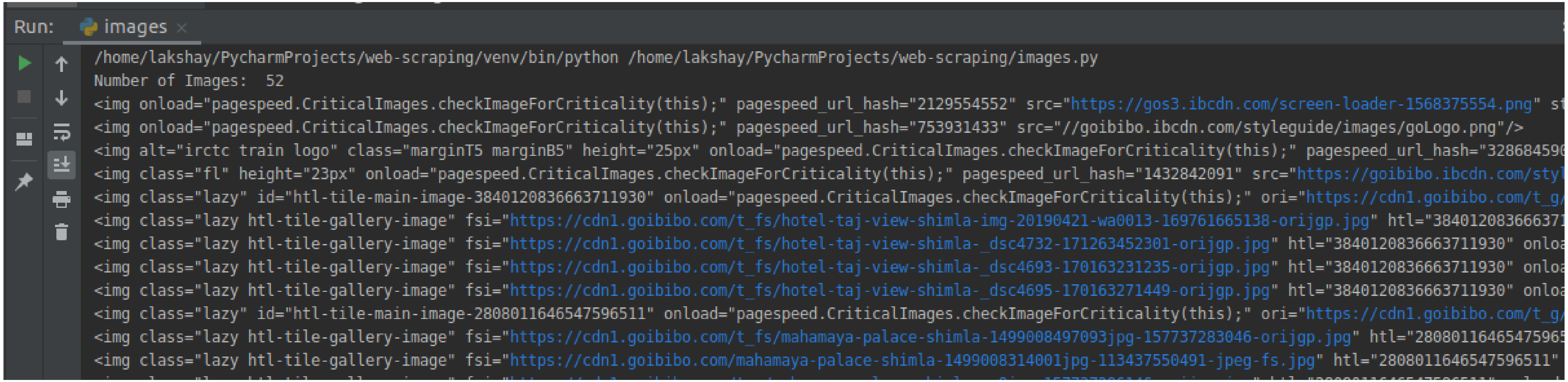
- In this section, we will scrape all the images from the same *goibibo* webpage. *goibibo.com/hotels/hotels-in-shimla-ct/*
- The first step would be same to navigate to the target website and download the source code.
- Next, we will find all the images using the **<img>** tag.
- To find all the images, we can use find_all() method

Find All Images



```
"""
Web Scraping - Scrap Images
"""

# importing required libraries
import requests
from bs4 import BeautifulSoup

# target URL
url = "https://www.goibibo.com/hotels/hotels-in-shimla-ct/"

headers = {
    'User-Agent': "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHT
    }

response = requests.request("GET", url, headers=headers)

data = BeautifulSoup(response.text, 'html.parser')

# find all with the image tag
images = data.find_all('img', src=True)

print('Number of Images: ', len(images))

for image in images:
    print(image)
```

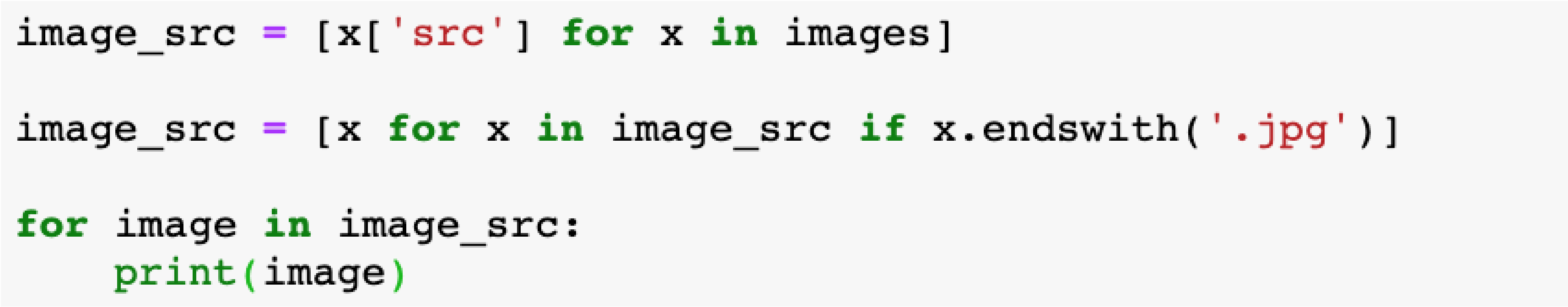
Result



```
/home/lakshay/PycharmProjects/web-scraping/venv/bin/python /home/lakshay/PycharmProjects/web-scraping/images.py
Number of Images:  52
<img onload="pagespeed.CriticalImages.checkImageForCriticality(this);" pagespeed_url_hash="2129554552" src="https://gos3.ibcdn.com/screen-loader-1568375554.png" st
<img onload="pagespeed.CriticalImages.checkImageForCriticality(this);" pagespeed_url_hash="753931433" src="//goibibo.ibcdn.com/styleguide/images/goLogo.png"/>
<img alt="irctc train logo" class="marginT5 marginB5" height="25px" onload="pagespeed.CriticalImages.checkImageForCriticality(this);" pagespeed_url_hash="32868459
<img class="fl" height="23px" onload="pagespeed.CriticalImages.checkImageForCriticality(this);" pagespeed_url_hash="1432842091" src="https://goibibo.ibcdn.com/styl
<img class="lazy" id="htl-tile-main-image-3840120836663711930" onload="pagespeed.CriticalImages.checkImageForCriticality(this);" ori="https://cdn1.goibibo.com/t_g/
<img class="lazy htl-tile-gallery-image" fsi="https://cdn1.goibibo.com/t_fs/hotel-taj-view-shimla-img-20190421-wa0013-169761665138-orijgp.jpg" htl="38401208366637
<img class="lazy htl-tile-gallery-image" fsi="https://cdn1.goibibo.com/t_fs/hotel-taj-view-shimla-_dsc4732-171263452301-orijgp.jpg" htl="3840120836663711930" onloa
<img class="lazy htl-tile-gallery-image" fsi="https://cdn1.goibibo.com/t_fs/hotel-taj-view-shimla-_dsc4693-170163231235-orijgp.jpg" htl="3840120836663711930" onloa
<img class="lazy htl-tile-gallery-image" fsi="https://cdn1.goibibo.com/t_fs/hotel-taj-view-shimla-_dsc4695-170163271449-orijgp.jpg" htl="3840120836663711930" onloa
<img class="lazy" id="htl-tile-main-image-2808011646547596511" onload="pagespeed.CriticalImages.checkImageForCriticality(this);" ori="https://cdn1.goibibo.com/t_g/
<img class="lazy htl-tile-gallery-image" fsi="https://cdn1.goibibo.com/t_fs/mahamaya-palace-shimla-1499008497093jpg-157737283046-orijgp.jpg" htl="280801164654759651
<img class="lazy htl-tile-gallery-image" fsi="https://cdn1.goibibo.com/mahamaya-palace-shimla-1499008314001jpg-113437550491-jpeg-fs.jpg" htl="2808011646547596511"
```

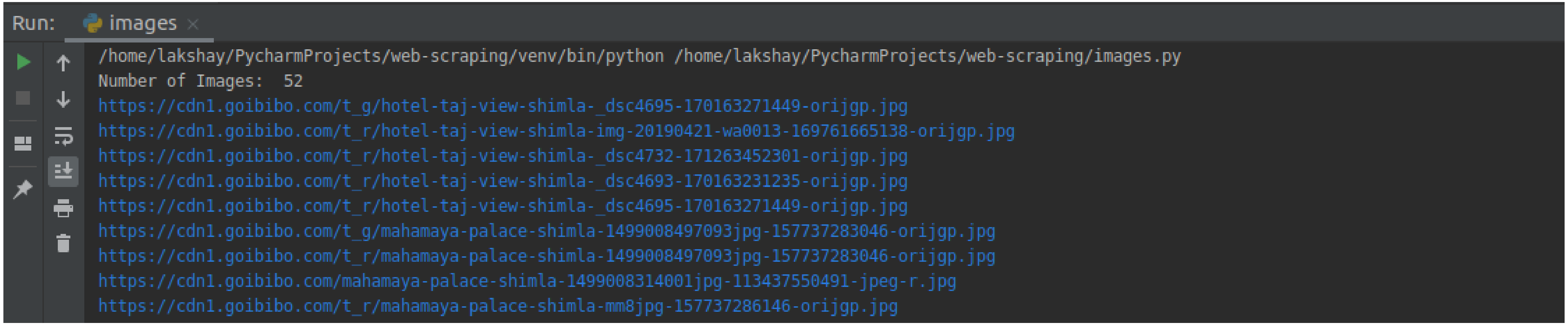
Get data image source

- From all the image tags, select only the ***src*** part. Also, notice that the hotel images are available in ***jpg*** format. So we will select only those



```
image_src = [x['src'] for x in images]

image_src = [x for x in image_src if x.endswith('.jpg')]

for image in image_src:
    print(image)
```

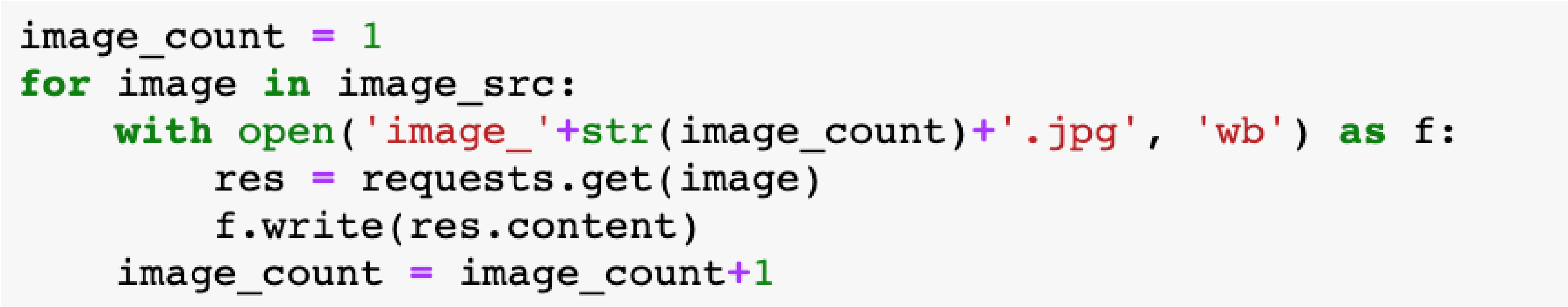
Result



```
/home/lakshay/PycharmProjects/web-scraping/venv/bin/python /home/lakshay/PycharmProjects/web-scraping/images.py
Number of Images:  52
https://cdn1.goibibo.com/t_g/hotel-taj-view-shimla-_dsc4695-170163271449-orijgp.jpg
https://cdn1.goibibo.com/t_r/hotel-taj-view-shimla-img-20190421-wa0013-169761665138-orijgp.jpg
https://cdn1.goibibo.com/t_r/hotel-taj-view-shimla-_dsc4732-171263452301-orijgp.jpg
https://cdn1.goibibo.com/t_r/hotel-taj-view-shimla-_dsc4693-170163231235-orijgp.jpg
https://cdn1.goibibo.com/t_r/hotel-taj-view-shimla-_dsc4695-170163271449-orijgp.jpg
https://cdn1.goibibo.com/t_g/mahamaya-palace-shimla-1499008497093jpg-157737283046-orijgp.jpg
https://cdn1.goibibo.com/t_r/mahamaya-palace-shimla-1499008497093jpg-157737283046-orijgp.jpg
https://cdn1.goibibo.com/mahamaya-palace-shimla-1499008314001jpg-113437550491-jpeg-r.jpg
https://cdn1.goibibo.com/t_r/mahamaya-palace-shimla-mm8jpg-157737286146-orijgp.jpg
```

Store Data

- Now that we have a list of image URLs, all we have to do is request the image content and write it in a file. Make sure that you open the file ***‘wb’*** (write binary) form:



```
image_count = 1
for image in image_src:
    with open('image_'+str(image_count)+'.jpg', 'wb') as f:
        res = requests.get(image)
        f.write(res.content)
    image_count = image_count+1
```

Result

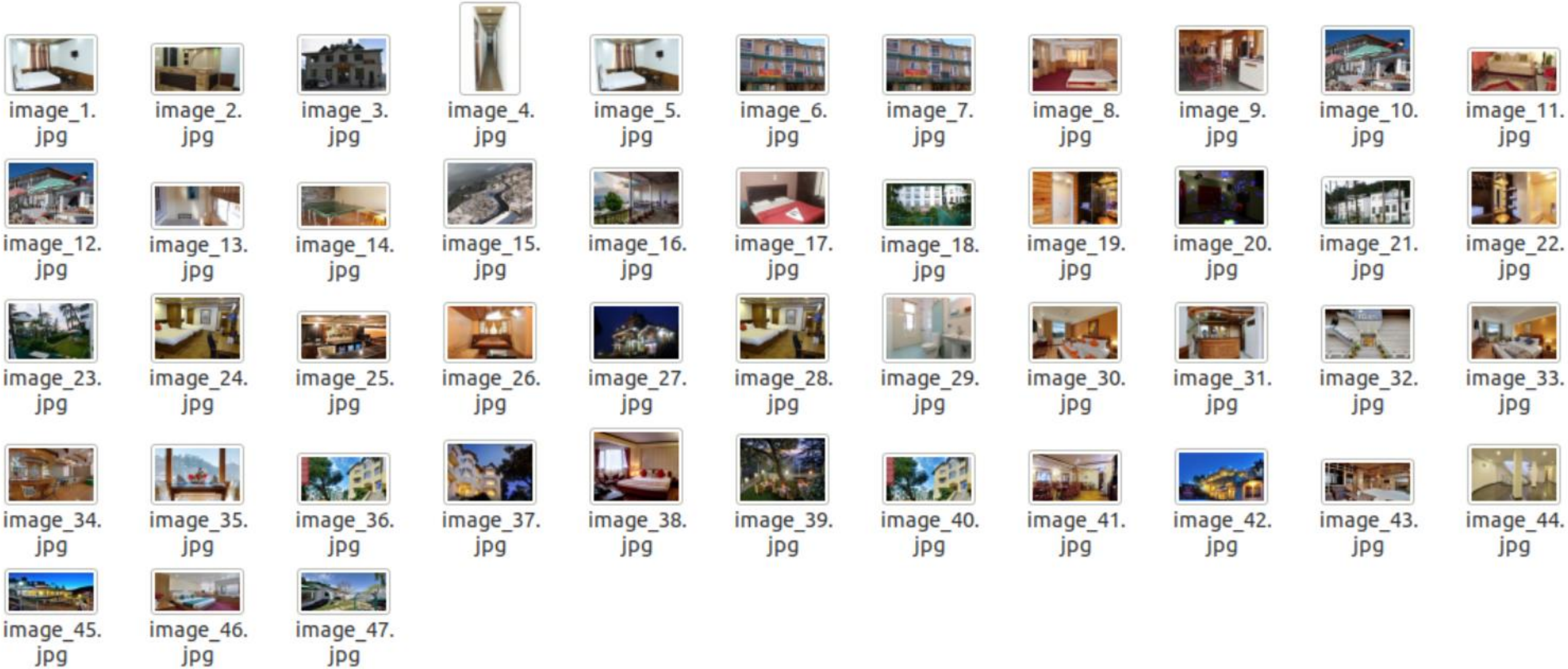


image_1.jpg
image_2.jpg
image_3.jpg
image_4.jpg
image_5.jpg
image_6.jpg
image_7.jpg
image_8.jpg
image_9.jpg
image_10.jpg
image_11.jpg
image_12.jpg
image_13.jpg
image_14.jpg
image_15.jpg
image_16.jpg
image_17.jpg
image_18.jpg
image_19.jpg
image_20.jpg
image_21.jpg
image_22.jpg
image_23.jpg
image_24.jpg
image_25.jpg
image_26.jpg
image_27.jpg
image_28.jpg
image_29.jpg
image_30.jpg
image_31.jpg
image_32.jpg
image_33.jpg
image_34.jpg
image_35.jpg
image_36.jpg
image_37.jpg
image_38.jpg
image_39.jpg
image_40.jpg
image_41.jpg
image_42.jpg
image_43.jpg
image_44.jpg
image_45.jpg
image_46.jpg
image_47.jpg

THANK YOU