

Ch-8 Web Programming Using Python(17 Marks)

web scraping using python(BeautifulSoup & requests)

- 9 Marks first code(Online Code)
- DeepLearning => 6 M theory code

```
In [2]: import requests
import bs4
url = "https://subslikescript.com/movie/Titanic-120338"
```

```
In [3]: source1 = requests.get(url)
print(source1)
```

<Response [200]>

```
In [4]: source = requests.get(url).text
print(source)
```

```

</div>
<nav>
    <a target="_self" href="/movies" title="All movies transcripts">Movies</a>
    <a target="_self" href="/series" title="All Tv Shows transcripts">TV Shows</a>
</nav>
</header>
<div class="main-wrapper">
    <main class="mainpage">
        <nav>
            <ul class="breadcrumb">
                <li><a href="/">Home</a></li>
                <li><a href="/movies">Movies</a></li>
                <li>Titanic - subtitles like script</li>
            </ul>
        </nav>
        <nav>
            <form class="searchzone" name="search" method="get" action="http
... /subslikescript.com/search">

```

```
In [5]: soup = bs4.BeautifulSoup(source, "html.parser")
print(soup)
```

```
<!DOCTYPE html>

<html dir="ltr" lang="en">
<head>
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-120598793-1"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'UA-120598793-1');
</script>
<meta charset="utf-8"/>
<title>    Titanic (1997) Movie Script
| Subs like Script</title>
<meta "" content=""    Read " created="" from="" movie="" name="description" script="" srt="" subtitles="" titanic="" />
```

```
In [6]: print(soup.prettify())
```

```
<!DOCTYPE html>
<html dir="ltr" lang="en">
<head>
  <!-- Global site tag (gtag.js) - Google Analytics -->
  <script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-120598793-1">
  </script>
  <script>
    window.dataLayer = window.dataLayer || [];
    function gtag(){dataLayer.push(arguments);}
    gtag('js', new Date());

    gtag('config', 'UA-120598793-1');
  </script>
  <meta charset="utf-8"/>
  <title>
    Titanic (1997) Movie Script
| Subs like Script
  </title>
```

```
In [7]: title = soup.find('h1').text
print(title)
```

Titanic (1997) - full transcript

```
In [8]: transcript = soup.find('p').text
print(transcript) # first p prefer.
```

A seventeen-year-old aristocrat falls in love with a kind but poor artist aboard the luxurious, ill-fated R.M.S. Titanic.

```
In [9]: transcript = soup.find('p',class_="cue-line").text  
print(transcript) # then only one line print because p tag is end.
```

13 meters. You should see it.

```
In [10]: transcript = soup.find('div',class_="full-script").text  
print(transcript) # full script mate parent tag no class Levano.
```

13 meters. You should see it.

Okay, take her up and over the bow rail.

Mir 2, we're going over the bow.
Stay with us.

Okay, quiet. We're rolling.

Seeing her coming out of the
darkness like a ghost ship...

... still gets me every time

```
In [11]: title = soup.find('h1').get_text()  
print(title)
```

Titanic (1997) - full transcript

```
In [12]: transcript = soup.find('div',class_="full-script").get_text()  
print(transcript)
```

13 meters. You should see it.

Okay, take her up and over the bow rail.

Mir 2, we're going over the bow.
Stay with us.

Okay, quiet. We're rolling.

Seeing her coming out of the
darkness like a ghost ship...

... still gets me every time

```
In [13]: f = open("Titanic.txt", "w", encoding="utf-8")
         f.write(transcript)
```

Out[13]: 67373

Task

```
In [14]: import requests
         import bs4
         url = "https://www.politifact.com/factchecks/"
```

```
In [15]: source1 = requests.get(url)
         source = requests.get(url).text
         soup = bs4.BeautifulSoup(source, "html.parser")
         transcript = soup.find('a', class_="m-statement__name").text
         print(transcript)
```

X posts

- find(string)=>text,get_text()
- find_all(list)

```
In [16]: # 1 mcq fix
         # transcript = soup.find_all('a', class_="m-statement__name").text
         # print(transcript)
```

```
In [17]: dic = {"statement":[], "source":[], "date":[]}
```

```
In [18]: transcript = soup.find_all('a',class_="m-statement__name")  
print(transcript)
```

```
[<a class="m-statement__name" href="/personalities/tweets/" title="X posts">
  X posts
</a>, <a class="m-statement__name" href="/personalities/tw
eets/" title="X posts">
  X posts
</a>, <a class="m-statement__name" href="/personalities/do
nald-trump/" title="Donald Trump">
  Donald Trump
</a>, <a class="m-statement__name" href="/personalities/tw
eets/" title="X posts">
  X posts
</a>, <a class="m-statement__name" href="/personalities/so
cial-media/" title="Social Media">
  Social Media
</a>, <a class="m-statement__name" href="/personalities/ga
vin-newsom/" title="Gavin Newsom">
  Gavin Newsom
</a>, <a class="m-statement__name" href="/personalities/ti
ktok-posts/" title="TikTok posts">
  TikTok posts
</a>, <a class="m-statement__name" href="/personalities/ka
roline-leavitt/" title="Karoline Leavitt">
  Karoline Leavitt
</a>, <a class="m-statement__name" href="/personalities/ro
bert-reich/" title="Robert Reich">
  Robert Reich
</a>, <a class="m-statement__name" href="/personalities/tw
eets/" title="X posts">
  X posts
</a>, <a class="m-statement__name" href="/personalities/sc
ott-bessent/" title="Scott Bessent">
  Scott Bessent
</a>, <a class="m-statement__name" href="/personalities/ro
bert-f-kennedy-jr/" title="Robert F. Kennedy Jr.">
  Robert F. Kennedy Jr.
</a>, <a class="m-statement__name" href="/personalities/be
rnie-sanders/" title="Bernie Sanders">
  Bernie Sanders
</a>, <a class="m-statement__name" href="/personalities/ga
vin-newsom/" title="Gavin Newsom">
  Gavin Newsom
</a>, <a class="m-statement__name" href="/personalities/do
nald-trump/" title="Donald Trump">
  Donald Trump
</a>, <a class="m-statement__name" href="/personalities/al
ex-jones/" title="Alex Jones">
  Alex Jones
</a>, <a class="m-statement__name" href="/personalities/ma
rcia-morey/" title="Marcia Morey">
  Marcia Morey
</a>, <a class="m-statement__name" href="/personalities/st
ephen-miller/" title="Stephen Miller">
  Stephen Miller
</a>, <a class="m-statement__name" href="/personalities/do
nald-trump/" title="Donald Trump">
  Donald Trump
</a>, <a class="m-statement__name" href="/personalities/do
nald-trump/" title="Donald Trump">
  Donald Trump
</a>, <a class="m-statement__name" href="/personalities/al
ex-jones/" title="Alex Jones">
```

```

        Alex Jones
    </a>, <a class="m-statement__name" href="/personalities/social-media/" title="Social Media">
        Social Media
    </a>, <a class="m-statement__name" href="/personalities/hakeem-jeffries/" title="Hakeem Jeffries">
        Hakeem Jeffries
    </a>, <a class="m-statement__name" href="/personalities/pam-bondi/" title="Pam Bondi">
        Pam Bondi
    </a>, <a class="m-statement__name" href="/personalities/charles-blow/" title="Charles Blow">
        Charles Blow
    </a>, <a class="m-statement__name" href="/personalities/tweets/" title="X posts">
        X posts
    </a>, <a class="m-statement__name" href="/personalities/tweets/" title="X posts">
        X posts
    </a>, <a class="m-statement__name" href="/personalities/donald-trump/" title="Donald Trump">
        Donald Trump
    </a>, <a class="m-statement__name" href="/personalities/donald-trump/" title="Donald Trump">
        Donald Trump
    </a>, <a class="m-statement__name" href="/personalities/jared-moskowitz/" title="Jared Moskowitz">
        Jared Moskowitz
    </a>]

```

```

In [19]: transcript = soup.find_all('a',class_="m-statement__name")
for i in transcript:
    b = i.text.strip()
    dic["source"].append(b)

```

```

In [20]: st = soup.find_all('div',class_="m-statement__quote")
print(st)

```

```

[<div class="m-statement__quote">
<a href="/factchecks/2025/may/23/tweets/LaMonica-McIver-CNN-altered-video-singing/">

        Rep. LaMonica McIver, D-N.J., started singing during a May 20 CNN interview.

    </a>
</div>, <div class="m-statement__quote">
<a href="/factchecks/2025/may/23/tweets/no-a-shooting-outside-the-capital-jewish-museum-wa/">

        The slaying of two Israeli embassy employees in Washington, D.C., is a false flag.

    </a>
</div>, <div class="m-statement__quote">
<a href="/factchecks/2025/may/22/donald-trump/medicaid-waste-fraud-abuse-changes/">

```

```
In [21]: st = soup.find_all('div',class_="m-statement__quote")
         for i in st:
             b = i.text.strip()
             dic["statement"].append(b)
```



```
In [22]: ft = soup.find_all('footer',class_="m-statement__footer")  
print(ft)
```

```
[<footer class="m-statement__footer">
  By Loreben Tuquero • May 23, 2025
</footer>, <footer class="m-statement__footer">
  By Maria Briceño • May 23, 2025
</footer>, <footer class="m-statement__footer">
  By Louis Jacobson • May 22, 2025
</footer>, <footer class="m-statement__footer">
  By Loreben Tuquero • May 22, 2025
</footer>, <footer class="m-statement__footer">
  By Grace Abels • May 21, 2025
</footer>, <footer class="m-statement__footer">
  By Amy Sherman • May 21, 2025
</footer>, <footer class="m-statement__footer">
  By Maria Briceño • May 21, 2025
</footer>, <footer class="m-statement__footer">
  By Louis Jacobson • May 21, 2025
</footer>, <footer class="m-statement__footer">
  By Loreben Tuquero • May 21, 2025
</footer>, <footer class="m-statement__footer">
  By Grace Abels • May 20, 2025
</footer>, <footer class="m-statement__footer">
  By Louis Jacobson • May 20, 2025
</footer>, <footer class="m-statement__footer">
  By Madison Czopek • May 16, 2025
</footer>, <footer class="m-statement__footer">
  By Louis Jacobson • May 16, 2025
</footer>, <footer class="m-statement__footer">
  By Amy Sherman • May 16, 2025
</footer>, <footer class="m-statement__footer">
  By Louis Jacobson • May 16, 2025
</footer>, <footer class="m-statement__footer">
  By Maria Briceño • May 16, 2025
</footer>, <footer class="m-statement__footer">
  By Paul Specht • May 15, 2025
</footer>, <footer class="m-statement__footer">
  By Maria Ramirez Uribe • May 15, 2025
</footer>, <footer class="m-statement__footer">
  By Louis Jacobson • May 14, 2025
</footer>, <footer class="m-statement__footer">
  By Amy Sherman • May 13, 2025
</footer>, <footer class="m-statement__footer">
  By Maria Briceño • May 13, 2025
</footer>, <footer class="m-statement__footer">
  By Maria Ramirez Uribe • May 12, 2025
</footer>, <footer class="m-statement__footer">
  By Louis Jacobson • May 12, 2025
</footer>, <footer class="m-statement__footer">
  By Maria Ramirez Uribe • May 9, 2025
</footer>, <footer class="m-statement__footer">
  By Loreben Tuquero • May 9, 2025
</footer>, <footer class="m-statement__footer">
  By Madison Czopek • May 9, 2025
</footer>, <footer class="m-statement__footer">
  By Grace Abels • May 9, 2025
</footer>, <footer class="m-statement__footer">
  By Madison Czopek • May 8, 2025
</footer>, <footer class="m-statement__footer">
  By Maria Ramirez Uribe • May 8, 2025
</footer>, <footer class="m-statement__footer">
```

```
In [23]: ft = soup.find_all('footer',class_="m-statement__footer")
for i in ft:
#     b = i.text.strip().split(".")[1]
    b = i.text[-31:-1].strip()
    dic["date"].append(b)
```

In [24]: `print(dic)`

```
{'statement': ['Rep. LaMonica McIver, D-N.J., started singing during a May 20 CNN interview.', 'The slaying of two Israeli embassy employees in Washington, D.C., is a false flag.', 'In the House bill, “we’re not changing Medicaid,” only cutting “waste, fraud and abuse.”', 'The Sean “Diddy” Combs trial testimony revealed that former President Barack Obama was “caught in secret meetings with Diddy’s drug runner.”', 'Video shows the Qatari emir saying, “I truly regret inviting Trump, the robber, to the Middle East.”', 'Que California ofrezca acceso asequible a cuidado de salud para los inmigrantes ilegalmente en EE.UU.', 'La presidenta de México Claudia Sheinbaum dijo que el choque del buque Cuauhtémoc contra el puente de Brooklyn en Nueva York no fue un accidente y puede ser un acto hostil.', 'President Donald Trump’s tax and spending bill “does not add to the deficit.”', 'While serving in Congress, Transportation Secretary Sean Duffy “voted against upgrading air traffic control systems.”', 'Former President Joe Biden’s cancer is a form of “turbo cancer” caused by mRNA vaccines.', '"Gasoline prices have collapsed under President Trump."', 'It’s “all true” that the measles vaccine wanes quickly, was never fully safety tested and contains fetal debris.', '"Republicans just unveiled their “big, beautiful bill,” which will take Medicaid & health insurance away from 13.7 million Americans."', 'On California offering affordable health care access for immigrants in the U.S. illegally.', 'Declining cargo traffic at U.S. ports “means we lose less money. ... When you say it's slowed down, that's a good thing.”', 'Un vídeo muestra al presidente francés, Emmanuel Macron, al primer ministro del Reino Unido, Keir Starmer, y al canciller alemán, Friedrich Merz, con una bolsa de cocaína tras visitar al presidente ucraniano, Volodymyr Zelenskyy.', '"Google caved in to Trump. They erased Black History Month from the calendar."', 'The Immigration and Nationality Act “stripped Article III courts, that's the judicial branch, of jurisdiction over immigration cases.”', 'Under a new executive order, prescription drug prices will be reduced “almost immediately.”', 'The Digital Equity Act is a handout “based on race,” and “illegal.”', 'A video shows French President Emmanuel Macron, United Kingdom Prime Minister Keir Starmer and German Chancellor Friedrich Merz with a bag of cocaine after visiting Ukrainian President Volodymyr Zelenskyy.', '"In 1996, Congress specifically authorized the executive branch to conduct non-judicial deportations NOT SUBJECT TO DUE PROCESS."', '"About 20% of households with veterans rely upon” the Supplemental Nutrition Assistance Program.', 'En los primeros 100 días del presidente Donald Trump, las incautaciones de fentanilo salvaron entre “119 millones” y “258 millones” de vidas.', '"China makes 80% of all toys” and “90% of all Christmas goods” sold in the U.S.', 'Headline said, “Pope Leo XIV was a founding member of the 2020 anarchist Portland Autonomous Zone known as CHAZ.”', 'Robert Prevost, the Roman Catholic cardinal who was chosen to be the new pope, is a “registered Republican.”', '"We have now close to $10 trillion” in investments. “We’re talking about essentially two months.”', '"If people come into our country illegally, there’s a different standard” for due process.', 'Canadian tourism to Florida has declined by 80%.'], 'source': ['X posts', 'X posts', 'Donald Trump', 'X posts', 'Social Media', 'Gavin Newsom', 'TikTok posts', 'Karoline Leavitt', 'Robert Reich', 'X posts', 'Scott Bessent', 'Robert F. Kennedy Jr.', 'Bernie Sanders', 'Gavin Newsom', 'Donald Trump', 'Alex Jones', 'Marcia Morey', 'Stephen Miller', 'Donald Trump', 'Donald Trump', 'Alex Jones', 'Social Media', 'Hakeem Jeffries', 'Pam Bondi', 'Charles Blow', 'X posts', 'X posts', 'Donald Trump', 'Donald Trump', 'Jared Moskowitz'], 'date': ['May 23, 2025', 'May 23, 2025', 'May 22, 2025', 'May 22, 2025', 'May 21, 2025', 'May 21, 2025', 'May 21, 2025', 'May 21, 2025', 'May 21, 2025', 'May 20, 2025', 'May 20, 2025', 'May 16, 2025', 'May 16, 2025', 'May 16, 2025', 'May 16, 2025', 'May 16, 2025', 'May 16, 2025', 'May 15, 2025', 'May 15, 2025', 'May 14, 2025', 'May 13, 2025', 'May 13, 2025', 'May 12, 2025', 'May 12, 2025', 'May 9, 2025', 'May 9, 2025', 'May 9, 2025', 'May 9, 2025', 'May 8, 2025', 'May 8, 2025', 'May 7, 2025']}]}
```

PB : 354

```
In [1]: import bs4
f = open("Today 52 Week Low BSE_NSE Stocks Companies List - Ticker.html", encoding="utf-8")
soup = bs4.BeautifulSoup(f, "html.parser")
print(soup.prettify())
```

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Global site tag (gtag.js) - Google Analytics -->
    <script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-136614031-6">
    </script>
    <script>
      window.dataLayer = window.dataLayer || [];
      function gtag() { dataLayer.push(arguments); }
      gtag('js', new Date());
      gtag('config', 'UA-136614031-6');
    </script>
    <title>
      Today 52 Week Low BSE/NSE Stocks Companies List - Ticker
    </title>
    <meta charset="utf-8"/>
    <meta content="width=device-width, initial-scale=1, shrink-to-fit=no" name="viewport"/>
```

```
In [56]: dic = {"sno.":[], "Company":[], "Price Rs : ":[], "Day low Rs.":[]}
```

Type Markdown and LaTeX: α^2

```
In [57]: com = soup.find_all('td', class_="left")
for i in com:
    b = i.text.strip()
    dic["Company"].append(b)
```

```
In [58]: num = soup.find_all('td', class_="Number")
for i in num:
    b = i.text.strip()
    dic["Price Rs : "].append(b)
```

```
In [59]: low = soup.find_all('span', class_="Number")
# sn=0
for i in low:
    b = i.text.strip()
    # sn+=1
    dic["Day low Rs."].append(b)
    # dic["sno."].append(sn)
```

In [60]: `print(dic)`

```
{'sno.': [], 'Company': ['Rachana Infrastru', 'Tirupati Forge', 'Sera Invest
ments&Fin', 'Viaz Tyres', 'Sellwin Traders', 'Soni Medicare', 'Sicagen Indi
a', 'Aspira Pathlab&Diagn', 'Patspin India', 'AG Universal', 'Arihant Found
n. &Hsg', 'Global Offshore Serv', 'GTN Textiles', 'GTN Inds', 'Vivanza Biosc
iences', 'TECIL Chem & Hydro', 'Milgreys Fin.&Invest', 'Zodiac-JRD-MKJ', 'Kan
ungo Financiers', 'CIL Nova Petro', 'Integ.Pro', 'Elango Inds', 'Voltaire Le
asing'], 'Price Rs. ': ['125.70', '7.80', '13.01', '46.35', '13.00', '19.2
2', '24.35', '24.85', '14.10', '42.10', '39.65', '7.85', '19.50', '11.95',
'8.33', '21.60', '16.29', '31.95', '5.35', '16.05', '8.81', '5.55', '11.9
9'], 'Day low Rs.': ['113.80', '7.75', '13.01', '45.00', '12.01', '17.40',
'24.00', '24.80', '14.00', '42.00', '38.50', '7.75', '19.50', '11.50', '8.3
3', '20.70', '14.81', '31.50', '4.85', '15.25', '8.81', '5.55', '11.54']}
```

In [50]: `sr = soup.find_all("tr")`
`print(sr)`

```
[<tr>
<th class="left" data-sortable="true" data-width="5%">S.No.</th>
<th class="left" data-sortable="true">Company</th>
<th data-sortable="true" data-sorter="changeSorter">price <span class="mut
ed">Rs.</span></th>
<th data-sortable="true" data-sorter="changeSorter">Day Low <span class="m
uted">Rs.</span></th>
</tr>, <tr>
<td>1</td>
<td class="left"><a href="/company/RILINFRA">Rachana Infrastru</a></td>
<td class="Number">125.70</td>
<td><span class="Number">113.80</span></td>
</tr>, <tr>
<td>2</td>
<td class="left"><a href="/company/TIRUPATIFL">Tirupati Forge</a></td>
<td class="Number">7.80</td>
<td><span class="Number">7.75</span></td>
</tr>, <tr>
<td>3</td>
<td class="left"><a href="/company/SCPTD_113300">Sera Investments&Fin</a></td>
```

```
In [67]: sr = soup.find_all("tr")
for i in sr[1:]:
    b=i.text[0:3].strip()
    dic["sno."].append(b)
print(dic)
```

```
{'sno.': ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23'], 'Company': ['Rachana Infrastru', 'Tirupati Forge', 'Sera Investments&Fin', 'Viaz Tyres', 'Sellwin Traders', 'Soni Medicare', 'Sicagen India', 'Aspira Pathlab&Diagn', 'Patspin India', 'AG Universal', 'Arihant Foundn. &Hsg', 'Global Offshore Serv', 'GTN Textile s', 'GTN Inds', 'Vivanza Biosciences', 'TECIL Chem & Hydro', 'Milgrey Fin.&I nvest', 'Zodiac-JRD-MKJ', 'Kanungo Financiers', 'CIL Nova Petro', 'Integ.Pr o', 'Elango Inds', 'Voltaire Leasing'], 'Price Rs : ': ['125.70', '7.80', '13.01', '46.35', '13.00', '19.22', '24.35', '24.85', '14.10', '42.10', '39.65', '7.85', '19.50', '11.95', '8.33', '21.60', '16.29', '31.95', '5.35', '16.05', '8.81', '5.55', '11.99'], 'Day low Rs.': ['113.80', '7.75', '13.01', '45.00', '12.01', '17.40', '24.00', '24.80', '14.00', '42.00', '38.50', '7.75', '19.50', '11.50', '8.33', '20.70', '14.81', '31.50', '4.85', '15.25', '8.81', '5.55', '11.54']}
```

```
In [66]: import pandas as pd
df = pd.DataFrame(dic)
print(df)
```

| | sno. | Company | Price Rs : | Day low Rs. |
|----|------|----------------------|------------|-------------|
| 0 | 1 | Rachana Infrastru | 125.70 | 113.80 |
| 1 | 2 | Tirupati Forge | 7.80 | 7.75 |
| 2 | 3 | Sera Investments&Fin | 13.01 | 13.01 |
| 3 | 4 | Viaz Tyres | 46.35 | 45.00 |
| 4 | 5 | Sellwin Traders | 13.00 | 12.01 |
| 5 | 6 | Soni Medicare | 19.22 | 17.40 |
| 6 | 7 | Sicagen India | 24.35 | 24.00 |
| 7 | 8 | Aspira Pathlab&Diagn | 24.85 | 24.80 |
| 8 | 9 | Patspin India | 14.10 | 14.00 |
| 9 | 10 | AG Universal | 42.10 | 42.00 |
| 10 | 11 | Arihant Foundn. &Hsg | 39.65 | 38.50 |
| 11 | 12 | Global Offshore Serv | 7.85 | 7.75 |
| 12 | 13 | GTN Textiles | 19.50 | 19.50 |
| 13 | 14 | GTN Inds | 11.95 | 11.50 |
| 14 | 15 | Vivanza Biosciences | 8.33 | 8.33 |
| 15 | 16 | TECIL Chem & Hydro | 21.60 | 20.70 |
| 16 | 17 | Milgrey Fin.&Invest | 16.29 | 14.81 |
| 17 | 18 | Zodiac-JRD-MKJ | 31.95 | 31.50 |
| 18 | 19 | Kanungo Financiers | 5.35 | 4.85 |
| 19 | 20 | CIL Nova Petro | 16.05 | 15.25 |
| 20 | 21 | Integ.Pro | 8.81 | 8.81 |
| 21 | 22 | Elango Inds | 5.55 | 5.55 |
| 22 | 23 | Voltaire Leasing | 11.99 | 11.54 |

PB : 351

```
In [84]: import bs4
f = open("Certified used Mercedes-Benz for sale.html",encoding="utf8")
soup = bs4.BeautifulSoup(f,"html.parser")
print(soup.prettify())

<!DOCTYPE html>
<html class="ep-theme-spark" lang="en">
  <head>
    <!-- s: no | l: no -->
    <base __cpgenerated="1" href="https://www.cars.com/shopping/results/?stock_type=cpo&makes%5B%5D=mercedes_benz&models%5B%5D=&list_price_max=&maximum_distance=20&zip="/>
    <!-- CP_UI_TAG -->
    <script __cpp="1">
      window.__Cpn = window.__Cpn ? window.__Cpn : function () { this.permalink = this.URI('https://stealthsurfer.online/__cpi.php?s=Q05rb0oxUjA5WmtybzZ6LzBIOGt0bEJaZzJBNOsvUXVhRElWSTBoRXNFenVlUFFpa01YN25NY1Z6b1lNqb1lYVU5KajgrSDVzZXNTTGy3OTc2cWRXMmZVemIzS0M5RFP6U015QnIvT21VUFk9&r=aHR0cHM6Ly9zdGVhbHRoc3VyZmVyLm9ubGluZS9zaG9wcGluZy9yZXN1bHRzLz9zdG9ja190eXB1PWNwbyZtYWtlcyU1QiU1RD1tZXJjZWRLc19iZW56Jm1vZGVscyU1QiU1RD0mbGlzdF9wcm1jZV9tYXg9Jm1heGltZW1fZGlzdGFuY2U9MjAmemlwPSZfX2Nwbz1hSFiwY0hNNkx5OTNkM2N1WTJGeWN5NWpiMjA%3D&cpo=1'); this.modal = '<style __cpp="1"> #__cpsModal #__cpsModalContent { background: url("...'); }'; }
    </script>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <h2>Certified used Mercedes-Benz for sale</h2>
          <div class="table">
            <table>
              <thead>
                <tr>
                  <th>Car Name</th>
                  <th>Mileage</th>
                  <th>Dealer Name</th>
                  <th>Review</th>
                  <th>Price</th>
                </tr>
              </thead>
              <tbody>
                <tr>
                  <td>Mercedes-Benz C-Class</td>
                  <td>12,000 miles</td>
                  <td>ABC Dealership</td>
                  <td>4.5</td>
                  <td>$25,000</td>
                </tr>
                <tr>
                  <td>Mercedes-Benz E-Class</td>
                  <td>18,000 miles</td>
                  <td>DEF Dealership</td>
                  <td>4.2</td>
                  <td>$35,000</td>
                </tr>
                <tr>
                  <td>Mercedes-Benz S-Class</td>
                  <td>25,000 miles</td>
                  <td>GHI Dealership</td>
                  <td>4.8</td>
                  <td>$55,000</td>
                </tr>
              </tbody>
            </table>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```
In [85]: dic = {"car Name":[],"mileage":[],"Dealer Name":[],"Review":[],"Price":[]}
```

```
In [86]: cn = soup.find_all("h2",class_="title")
for i in cn:
    b = i.text.strip()
    dic["car Name"].append(b)
```

```
In [87]: ml = soup.find_all("div",class_="mileage")
for i in ml[1:]:
    b = i.text.strip()
    dic["mileage"].append(b)
```

```
In [88]: dn = soup.find_all("div",class_="dealer-name")
for i in dn:
    b = i.text.strip()
    dic["Dealer Name"].append(b)
```

```
In [89]: re = soup.find_all("span",class_="sds-rating__link sds-button-link")
for i in re:
    b = i.text.strip("()reviews")
    dic["Review"].append(b)
```

```
In [90]: pr = soup.find_all("span",class_="primary-price")
for i in pr:
    b = i.text.strip()
    dic["Price"].append(b)
```



```
In [91]: import pandas as pd
df = pd.DataFrame(dic)
df.head()
```

```
Out[91]:
```

| | car Name | mileage | Dealer Name | Review | Price |
|---|---|------------|-----------------------------|--------|----------|
| 0 | 2020 Mercedes-Benz CLA 250 Base 4MATIC | 28,744 mi. | Mercedes-Benz of Lynnwood | 130 | \$35,995 |
| 1 | 2019 Mercedes-Benz AMG GT 53 Base | 25,771 mi. | International Autos | 117 | \$79,995 |
| 2 | 2021 Mercedes-Benz AMG GLE 53 Base | 22,374 mi. | John Sisson Motors | 38 | \$80,923 |
| 3 | 2022 Mercedes-Benz AMG CLA 45 Base 4MATIC | 10,595 mi. | Mercedes-Benz of Santa Rosa | 30 | \$56,633 |
| 4 | 2021 Mercedes-Benz AMG GLE 53 Base | 33,622 mi. | Mercedes-Benz of Rochester | 152 | \$76,995 |

- API KEY :669ccb11b84117c0c3ed6551cebb8d20

Application Programming Interface(API)(Geocode API)

```
In [2]: import requests,json
def get_coordinates():
    api_key = "669ccb11b84117c0c3ed6551cebb8d20"
    city = "Surat"
    url = "http://api.openweathermap.org/geo/1.0/direct?q="+city+"&appid="+api_key
    response = requests.get(url).json()
    print(json.dumps(response,indent=5)) # with Proper Indentatation
    latitude = response[0]["lat"]
    longitude = response[0]["lon"]
    return latitude,longitude
lat,long=get_coordinates()
```

```
[
  {
    "name": "Surat",
    "local_names": {
      "fr": "Surat",
      "uk": "\u0421\u0443\u0440\u0430\u0442"
    },
    "lat": 45.9383,
    "lon": 3.2553,
    "country": "FR",
    "state": "Auvergne-Rh\u00f4ne-Alpes"
  }
]
```

```
In [5]: # Current Data API
import requests,json
def get_coordinates():
    api_key = "669ccb11b84117c0c3ed6551cebb8d20"
    city = "Surat"
    url = "https://api.openweathermap.org/data/2.5/weather?lat="+str(lat)+"&lc
    response = requests.get(url).json()
    print(json.dumps(response,indent=5))
    tmp = response["main"]["temp"]
    pres = response["main"]["pressure"]
    hum = response["main"]["humidity"]
    des = response["weather"][0]["description"]
    ws = response["wind"]["speed"]
    vt = response["visibility"]
    return tmp,pres,hum,des,ws,vt
tmp,pres,hum,des,ws,vt=get_coordinates()
```

```
In [6]: # Air Pollution API
import requests, json

def get_coordinates():
    api_key="669ccb11b84117c0c3ed6551cebb8d20"
    city="Surat"
    url = "https://api.openweathermap.org/data/2.5/air_pollution?lat="+str(lat)
    response=requests.get(url).json()
    print(json.dumps(response,indent=5))
    aqi = response["list"][0]["main"]["aqi"]
    return aqi
aqi = get_coorinates()
d = {1:"excellent",2:"Good",3:"Average",4:"poor",5:"verybad"}
print(d[aqi])

# Output:
# {
#   "coord": {
#     "lon": 3.2553,
#     "lat": 45.9383
#   },
#   "list": [
#     {
#       "main": {
#         "aqi": 1
#       },
#       "components": {
#         "co": 84.34,
#         "no": 1.35,
#         "no2": 1.96,
#         "o3": 33.02,
#         "so2": 0.1,
#         "pm2_5": 1.65,
#         "pm10": 2.17,
#         "nh3": 19.22
#       },
#       "dt": 1748590421
#     }
#   ]
# }
# excellent
```

```
In [ ]: import requests, json

def get_coordinates():
    api_key="669ccb11b84117c0c3ed6551cebb8d20"
    city="Surat"
    url = "https://api.openweathermap.org/data/2.5/forecast?lat="+str(lat)+"&I
    response=requests.get(url).json()
    print(json.dumps(response,indent=5))
    return response
five_day = five_day_forecast()
D = {"data_time":[],"temp":[],"Pressure":[],"humidity":[],"Description":[]}
for i in five_day["list"]:
    D["date_time"].append(i["dt_text"])
    D["temp"].append(i["main"]["temp"])
    D["Pressure"].append(i["main"]["pressure"])
    D["humidity"].append(i["main"]["humidity"])
    D["Description"].append(i["weather"][0]["description"])
```

```
In [ ]: import pandas as pd
df = pd.DataFrame(D)
df
```