

[15]: [I write a python program to display all the header tags from wikipedia.org.](#)

```
import requests
from bs4 import BeautifulSoup
url_link = 'https://en.wikipedia.org/wiki/Main_Page'
request = requests.get(url_link)
soup = BeautifulSoup(request.text, 'lxml')
#Extracting list of all common heading tags
heading_tags = ["h1", "h2", "h3"]
for tags in soup.find_all(heading_tags):
    print(tags.name + " -> " + tags.text.strip())

# Main Page
# -> Welcome to Wikipedia
# -> Get today's featured article
# -> Did you know ...
# -> On this day
# -> From today's featured list
# -> Today's featured picture
# -> Wikipedia's sister projects
# -> Wikipedia languages
# -> Navigation menu
# -> Personal tools
# -> Namespaces
# -> Views
# -> Search
# -> Navigation
# -> Tools
# -> Print/export
# -> In other projects
# -> Languages
```

```
In [10]: !pip install bs4
import requests
#Requirement already satisfied: BeautifulSoup in c:\programdata\anaconda3\lib\site-packages (from bs4) (4.9.3)
#Requirement already satisfied: soupsieve>1.2; python_version >= '3.0' in c:\programdata\anaconda3\lib\site-packages (from BeautifulSoup->bs4) (2.0.0)
#Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (from requests) (2.24.0)
#Requirement already satisfied: urllib3<1.25.0, >=1.25.0; python_version < '3.10' in c:\programdata\anaconda3\lib\site-packages (from requests) (1.26.13)
#Requirement already satisfied: charset-normalizer<4, >=3.2 in c:\programdata\anaconda3\lib\site-packages (from requests) (3.4.4)
#Requirement already satisfied: certifi<2024.4.17, >=2024.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests) (2024.6.20)
#Requirement already satisfied: idna<4, >=3 in c:\programdata\anaconda3\lib\site-packages (from requests) (2.10)

In [42]: #Write a python program to display IMDb's Top rated 100 movies' data (i.e. name, rating, year of release)and make data frame.
from bs4 import BeautifulSoup
request = requests.get('https://www.imdb.com/search/title/?groups=top_100&sort=user_rating_desc')
soup = BeautifulSoup(page.content)
movie_name = []
for i in soup.find_all('h3', class_='list-item-header'):
    print(movie_name.append(i.text.replace("\n", "")))
    movie_name
rating = []
for i in soup.find_all('div', class_='inline-block ratings-imdb-rating'):
    rating.append(i.text.replace("\n", ""))
rating
year = []
for i in soup.find_all('span', class_='list-item-year text-muted unbold'):
    year.append(i.text)
year

import pandas as pd
df = pd.DataFrame({'movie_name': movie_name, 'rating': rating, 'year': year})
df
```

```
Out[42]:
```

	movie_name	rating	year
0	1.The Shawshank Redemption(1994)	9.3	(1994)
1	2.The Godfather(1972)	9.2	(1972)
2	3.The Dark Knight(2008)	9.0	(2008)
3	4.The Lord of the Rings: The Return of the Kin...	9.0	(2003)
4	5.Schindler's List(1993)	9.0	(1993)
5	6.The Godfather Part II(1974)	9.0	(1974)
6	7.12 Angry Men(1957)	9.0	(1957)
7	8.Pulp Fiction(1994)	8.9	(1994)
8	9.Inception(2010)	8.8	(2010)
9	10.The Lord of the Rings: The Two Towers(2002)	8.8	(2002)
10	11.Fight Club(1999)	8.8	(1999)
11	12.The Lord of the Rings: The Fellowship of th...	8.8	(2001)
12	13.Forest Gump(1994)	8.8	(1994)
13	14.El bueno, li bruto, i cattivo(1966)	8.8	(1966)
14	15.The Matrix(1999)	8.7	(1999)
15	16.Godfather(1980)	8.7	(1980)
16	17.The Empire Strikes Back(1980)	8.7	(1980)
17	18.One Flew Over the Cuckoo's Nest(1975)	8.7	(1975)
18	19.Top Gun: Maverick(2022)	8.6	(2022)
19	20.Stand by Me(1986)	8.6	(1986)
20	21.Casino de Paris(2004)	8.6	(2004)
21	22.Sen to Chiroro no kamikakushi(2001)	8.6	(2001)
22	23.Saving Private Ryan(1998)	8.6	(1998)
23	24.The Green Mile(1999)	8.6	(1999)
24	25.La vie e bella(1997)	8.6	(1997)
25	26.Selena(1995)	8.6	(1995)
26	27.Terminator 2: Judgment Day(1991)	8.6	(1991)
27	28.The Silence of the Lambs(1991)	8.6	(1991)
28	29.Six Years(1977)	8.6	(1977)
29	30.Spansky(1962)	8.6	(1962)
30	31.Shichirin no samurai(1954)	8.6	(1954)
31	32.It's a Wonderful Life(1946)	8.6	(1946)
32	33.Guangyong(2009)	8.5	(2009)
33	34.Wings(2004)	8.5	(2004)
34	35.The Invention(2011)	8.5	(2011)
35	36.The Pianist(2002)	8.5	(2002)
36	37.The Departed(2006)	8.5	(2006)
37	38.The Pianist(2002)	8.5	(2002)
38	39.Gladstone(2000)	8.5	(2000)
39	40.American History X(1998)	8.5	(1998)
40	41.The Usual Suspects(1995)	8.5	(1995)
41	42.Lenny(1994)	8.5	(1994)
42	43.The Lion King(1994)	8.5	(1994)
43	44.Nuovo Cinema Paradiso(1988)	8.5	(1988)
44	45.Hotaru no haka(1988)	8.5	(1988)
45	46.Back to the Future(1985)	8.5	(1985)
46	47.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
47	48.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
48	49.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
49	50.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
50	51.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
51	52.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
52	53.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
53	54.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
54	55.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
55	56.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
56	57.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
57	58.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
58	59.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
59	60.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
60	61.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
61	62.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
62	63.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
63	64.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
64	65.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
65	66.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
66	67.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
67	68.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
68	69.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
69	70.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
70	71.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
71	72.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
72	73.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
73	74.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
74	75.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
75	76.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
76	77.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
77	78.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
78	79.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
79	80.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
80	81.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
81	82.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
82	83.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
83	84.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
84	85.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
85	86.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
86	87.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
87	88.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
88	89.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
89	90.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
90	91.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
91	92.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
92	93.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
93	94.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
94	95.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
95	96.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
96	97.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
97	98.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
98	99.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
99	100.A.P. Nagaraju (1997-2009)Term of Off...	8.5	(1997)
100 rows x 3 columns			

```
In [43]: #Write a python program to display IMDb's Top rated 100 Indian movies' data (i.e. name, rating, year of release) and make data frame.
from bs4 import BeautifulSoup
request = requests.get('https://www.imdb.com/list/ls000997493/')
page = requests.get('https://www.imdb.com/list/ls000997493/')
soup = BeautifulSoup(page.content)
movie_name = []
for i in soup.find_all('h3', class_='list-item-header'):
    print(movie_name.append(i.text.replace("\n", "")))
    movie_name
rating = []
for i in soup.find_all('div', class_='ipl-rating-star small'):
    rating.append(i.text.replace("\n", ""))
rating
year = []
for i in soup.find_all('span', class_='list-item-year text-muted unbold'):
    year.append(i.text)
year

import pandas as pd
df = pd.DataFrame({'movie_name': movie_name, 'rating': rating, 'year': year})
df
```

```
Out[43]:
```

	movie_name	rating	year
0	1.Rang De Basanti(2006)	8.1	(2006)
1	2.3 Idiots(2009)	8.4	(2009)
2	3.Taare Zameen Par(2007)	8.3	(2007)
3	4.Dil Chhata Hai(2001)	8.1	(2001)
4	5.Swades: We, the People(2004)	8.1	(2004)
...
96	96.Waale Waad(2009)	7.6	(2009)
97	97.Rang De Basanti(2006)	7.4	(2006)
98	98.Waale Waad(2009)	7.5	(2009)
99	99.Yaar Ka Parchayam(2011)	7.6	(2011)
100	100.Ha Ha Ha(2004)	7.5	(2004)
100 rows x 3 columns			

```
In [44]: #Write a python program to display list of respected former presidents of India
from bs4 import BeautifulSoup
request = requests.get('https://presidentofindia.nic.in/former-presidents.htm')
page = requests.get('https://presidentofindia.nic.in/former-presidents.htm')
soup = BeautifulSoup(page.content)
name = []
for i in soup.find_all('div', class_='presidentListing'):
    print(name.append(i.text.replace("\n", "")))
    name
import pandas as pd
df = pd.DataFrame({'name': name})
df
```

```
Out[44]:
```

	name
0	Shri Ram Nath Kovind(Born:1945)Term of Off...
1	Shri Pranab Mukherjee (1915-2020)Term of Off...
2	Sri Pratibha Devi Singh Patel (Born: 1934)Ter...
3	DR. A.P.J. Abdul Kalam (1931-2015)Term of Off...
4	Sri K.R. Narayanan (1902-2009)Term of Off...
5	Shri Shankar Dayal Sharma (1918-2007)Term of Off...
6	Shri B.V. Kesavaiah (1905-2009)Term of Off...
7	Shri Zail Singh (1916-1994)Term of Office: 25...
8	Shri Neelam Sanjiva Reddy (1913-1996)Term of O...
9	Dr. Fakrudin Ali Ahmed (1905-1977)Term of Off...
10	Shri Venkatesh Swamy (1904-1980)Term of Off...
11	Dr. Zakir Husain (1897-1967)Term of Office: 13...
12	Dr. Sarvepalli Radhakrishnan (1888-1975)Term o...
13	Dr. Rajendra Prasad (1884-1963)Term of Office...

```
In [19]: from urllib.request import urlopen
from bs4 import BeautifulSoup as Soup
import requests as rs
page_OOI_Bat = rs.get('https://www.icc-cricket.com/rankings/mens/team-rankings/odi')
OOI_soup_Bat = Soup(page_OOI_Bat.content, "html.parser")
Batmen_Name = []
Team = []
Batmen = OOI_soup_Bat.find_all("td", class_="table-body_cell name")
Batmen[0:4]
for i in Batmen:
    Batmen_soup.append(i.get_text().replace("\n", ""))
    Batmen

Team[0:3]
TeamNames = OOI_soup_Bat.find_all("span", class_="table-body_logo-text")
TeamNames[0:4]
for i in TeamNames:
    Team.append(i.get_text().replace("\n", ""))
    Team

Team[0:4]
Rating_Bat = OOI_soup_Bat.find_all("td", class_="table-body_cell u-text-right rating")
Rating_Bat[0:4]
for i in Rating_Bat:
    Rating.append(i.get_text().replace("\n", ""))
    Rating

Rating[0:4]
Batmen_Name.insert(0, 'Babar Azam')
Team.insert(0, 'PAK')
Rating.insert(0, '815')
Batmen.insert(0, '815')
import pandas as pd
Top_Batmen_OOI = pd.DataFrame({})
Top_Batmen_OOI['Batmen_Names'] = Batmen_Name
Top_Batmen_OOI['Team_Names'] = Team
Top_Batmen_OOI['Rating'] = Rating
Top_Batmen_OOI.head(10)
```

```
Out[19]:
```

	Batmen_Names	Team_Names	Rating
0	Babar Azam	PAK	815
1	Ishan-Ki-Hag	PAK	815
2	Rassie van der Dussen	SA	789
3	Quinton de Kock	SA	784
4	Virat Kohli	IND	767
5	Rohit Sharma	IND	763
6	Ross Taylor	NZ	744
7	David Warner	AUS	737
8	Jonny Bairstow	ENG	732
9	Aaron Finch	AUS	715

```
In [22]: from urllib.request import urlopen
from bs4 import BeautifulSoup as Soup
import requests as rs
page_OOI_Bow = rs.get('https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling')
OOI_soup_Bow = Soup(page_OOI_Bow.content, "html.parser")
Bowling_Name = []
Teams = []
Bowling = OOI_soup_Bow.find_all("td", class_="table-body_cell rankings-table_name name")
Bowling[0:4]
for i in Bowling:
    Bowling_Name.append(i.get_text().replace("\n", ""))
    Bowling

Bowling_Name[0:3]
TeamNames = OOI_soup_Bow.find_all("span", class_="table-body_logo-text")
TeamNames[0:4]
for i in TeamNames:
    Teams.append(i.get_text().replace("\n", ""))
    Teams

Teams[0:4]
Rating_Bows = OOI_soup_Bow.find_all("td", class_="table-body_cell rating")
Rating_Bows[0:4]
for i in Rating_Bows:
    Ratings.append(i.get_text().replace("\n", ""))
    Ratings

Ratings[0:4]
Bowling_Name.insert(0, 'Trent Boult')
Teams.insert(0, 'NZ')
Ratings.insert(0, '697')
import pandas as pd
Top_Bowling_OOI = pd.DataFrame({})
Top_Bowling_OOI['Bowling_Names'] = Bowling_Name
Top_Bowling_OOI['Team_Names'] = Teams
Top_Bowling_OOI['Rating'] = Ratings
Top_Bowling_OOI.head(10)
```

```
Out[22]:
```

	Batmen_Names	Team_Names	Rating
0	Trent Boult	NZ	697
1	Jasprit Bumrah	IND	682
2	Shaheen Afridi	PAK	681
3	Josh Hazlewood	AUS	679
4	Mageul ur Rahman	AFG	676
5	Mehedi Hasan	BAN	672
6	Matt Henry	NZ	663
7	Muhammad Nabi	AFG	657
8	Rashid Khan	AFG	651
9	Chris Woakes	ENG	640

```
In [23]: from urllib.request import urlopen
from bs4 import BeautifulSoup as Soup
import requests as rs
page_OOI_W = rs.get('https://www.icc-cricket.com/rankings/womens/team-rankings/odi')
OOI_soup_W = Soup(page_OOI_W.content, "html.parser")
Team = []
Match = []
Points = []
Rating = []
W_Team = OOI_soup_W.find_all("span", class_="u-hide-phabiet")
W_Team[0:4]
for i in W_Team:
    Team.append(i.get_text().replace("\n", ""))
    Team

Team[0:10]
OOI_Match_W = OOI_soup_W.find_all("td", class_="table-body_cell u-center-text")
OOI_Match_W[0:4]
for i in OOI_Match_W:
    Match.append(i.get_text().replace("\n", ""))
    Match

Match[0:4]
Total_Match = []
i = 0
while i < len(Match):
    Total_Match.append(Match[i])
    i += 2
print(Total_Match)
Points = []
i = 0
while i < len(Match):
    Points.append(Match[i])
    i += 2
print(Points)
OOI_Rating_W = OOI_soup_W.find_all("td", class_="table-body_cell u-text-right rating")
OOI_Rating_W[0:4]
for i in OOI_Rating_W:
    Rating.append(i.get_text().replace("\n", ""))
    Rating

Rating[0:4]
import pandas as pd
Top_10_Monan = pd.DataFrame({})
Top_10_Monan['Team_Name'] = Team
Top_10_Monan['Total_Match'] = Total_Match
Top_10_Monan['Points'] = Points
Top_10_Monan['Rating'] = Rating
Top_10_Monan.head(10)
```

```
Out[23]:
```

	Team_Name	Total_Match	Points	Rating
0	Australia	33	4086	123
1	England	35	4157	119
2	South Africa	32	3219	101
3	India	31	3019	97
4	New Zealand	30	2768	92
5	West Indies	12	930	78
6	Bangladesh	30	1362	65
7	Pakistan	11	495	45
8	Sri Lanka	8	351	44
9	Ireland	8	0	0

```
In [24]: from urllib.request import urlopen
from bs4 import BeautifulSoup as Soup
import requests as rs
page_OOI_Bat_W = rs.get('https://www.icc-cricket.com/rankings/womens/player-rankings/odi')
OOI_soup_Bat_W = Soup(page_OOI_Bat_W.content, "html.parser")
Player_Name = []
Teams = []
Player_W = OOI_soup_Bat_W.find_all("td", class_="table-body_cell name")
Player_W[0:4]
for i in Player_W:
    Player_Name.append(i.get_text().replace("\n", ""))
    Player_Name

Player_Name[0:3]
Team_Name_W = OOI_soup_Bat_W.find_all("span", class_="table-body_logo-text")
Team_Name_W[0:4]
for i in Team_Name_W:
    Teams.append(i.get_text().replace("\n", ""))
    Teams

Teams[0:4]
Rating_Bows_W = OOI_soup_Bat_W.find_all("td", class_="table-body_cell u-text-right rating")
Rating_Bows_W[0:4]
for i in Rating_Bows_W:
    Rating_W.append(i.get_text().replace("\n", ""))
    Rating_W

Rating_W[0:4]
Player_Name.insert(0, 'Alyssa Healy')
Teams.insert(0, 'AUS')
Rating_W.insert(0, '780')
print(len(Team_W), len(Player_Name), len(Rating_W))
import pandas as pd
Top_Player_OOI_W = pd.DataFrame({})
Top_Player_OOI_W['Player_Name'] = Player_Name
Top_Player_OOI_W['Team_Names'] = Team_Name
Top_Player_OOI_W['Rating'] = Rating_W
Top_Player_OOI_W.head(10)
```

```
Out[24]:
```

	Batmen_Names	Team_Names	Rating
0	Alyssa Healy	AUS	785
1	Beth Mooney	AUS	749
2	Natalie Sciver	ENG	747
3	Lauren Winward	SA	732
4	Meg Lanning	AUS	710
5	Rachael Haynes	AUS	704
6	Amy Satterthwaite	NZ	681
7	Ellyse Baillie	ENG	667
8	Chloe Athey	AUS	655
9	Sarah Mowbray	IND	649

```
In [25]: from urllib.request import urlopen
from bs4 import BeautifulSoup as Soup
import requests as rs
page_OOI_All = rs.get('https://www.icc-cricket.com/rankings/womens/player-rankings/odi/all-rounder')
OOI_soup_All = Soup(page_OOI_All.content, "html.parser")
All_Rounder_Name = []
Team_All = []
Rating_All = []
All_Rounder = OOI_soup_All.find_all("td", class_="table-body_cell rankings-table_name name")
All_Rounder[0:4]
for i in All_Rounder:
    All_Rounder_Name.append(i.get_text().replace("\n", ""))
    All_Rounder_Name

All_Rounder_Name[0:3]
Team_Name_All = OOI_soup_All.find_all("span", class_="table-body_logo-text")
Team_Name_All[0:4]
for i in Team_Name_All:
    Team_All.append(i.get_text().replace("\n", ""))
    Team_All

Team_All[0:4]
Team_Name_All = OOI_soup_All.find_all("span", class_="table-body_logo-text")
Team_Name_All[0:4]
for i in Team_Name_All:
    Team_All.append(i.get_text().replace("\n", ""))
    Team_All

Team_All[0:4]
Rating_All_W = OOI_soup_All.find_all("td", class_="table-body_cell rating")
Rating_All_W[0:4]
for i in Rating_All_W:
    Rating_All.append(i.get_text().replace("\n", ""))
    Rating_All

Rating_All[0:4]
All_Rounder_Name.insert(0, 'Natalie Sciver')
Team_All.insert(0, 'ENG')
Rating_All.insert(0, '379')
import pandas as pd
All_OOI_W = pd.DataFrame({})
All_OOI_W['All_Rounder_Name'] = All_Rounder_Name
All_OOI_W['Team_Names'] = Team_All[0:20]
All_OOI_W['Rating'] = Rating_All
All_OOI_W.head(10)
```

```
Out[25]:
```

	All_Rounder_Name	Team_Names	Rating
0	Natalie Sciver	ENG	379
1	Ellyse Perry	AUS	374
2	Maryanne Kapp	SA	349
3	Hayley Matthews	WI	339
4	Aneesh Chandra	NZ	339
5	Ashleigh Gardner	AUS	270
6	Deepti Sharma	IND	252
7	Jess Jonsson	AUS	246
8	Katherine Brunst	ENG	220
9	Stefanie Taylor	WI	207

```
In [26]: from bs4 import BeautifulSoup
import requests
#https://www.dineout.co.in/delhi-restaurants/buffet-special
soup = BeautifulSoup(soup.find('a', class_="restnt-name ellipsis"))
cuisine = soup.find('div', class_="double-line-ellipsis")
for i in soup.find_all('span', class_="double-line-ellipsis"):
    location = soup.find('div', class_="restnt-loc ellipsis")
    location_text = soup.find('div', class_="restnt-rating rating-4")
    ratings = soup.find('div', class_="no-img")
    for i in soup.find_all('img', class_="data-src"):
        images.append(i['data-src'])
    titles = []
    for i in soup.find_all('a', class_="restnt-name ellipsis"):
        titles.append(i.text)
    titles
    titles
    for i in soup.find_all('div', class_="restnt-loc ellipsis"):
        location.append(i.text)
    ratings = []
    for i in soup.find_all('div', class_="restnt-rating rating-4"):
        ratings.append(i.text)
    ratings
    import pandas as pd
    df = pd.DataFrame({'titles': titles, 'cuisine': cuisine, 'location': location, 'ratings': ratings, 'images': images})
    df
```

```
Out[26]:
```

	titles	cuisine	location	ratings	images
0	Casde Bellevue	<2			