1)ALL HEADER TAG FROM WIKIPEDIA

```
In [78]: from bs4 import BeautifulSoup
```

```
In [79]: import requests
```

```
In [80]: page=requests.get("https://en.wikipedia.org/wiki/Main_Page")
```

```
In [81]: page
```

```
Out[81]: <Response [200]>
```

```
In [82]: soup=BeautifulSoup(page.text,'html.parser')
```

```
In [83]: print(soup.prettify())
```

```
...
```

```
In [84]: headings=soup.find_all(['h1','h2','h3','h4','h5','h6'])
```

In [85]:

```python
for i in headings:
    print(i)
```

```
<h1 class="firstHeading" id="firstHeading">Main Page</h1>
<h2 class="mp-h2" id="mp-tfa-h2"><span id="From_today.27s_featured_article"></s
pan><span class="mw-headline" id="From_today's_featured_article">From today's f
eatured article</span></h2>
<h2 class="mp-h2" id="mp-dyk-h2"><span class="mw-headline" id="Did_you_know
_...">Did you know ...</span></h2>
<h2 class="mp-h2" id="mp-itn-h2"><span class="mw-headline" id="In_the_news">In
the news</span></h2>
<h2 class="mp-h2" id="mp-otd-h2"><span class="mw-headline" id="On_this_day">On
this day</span></h2>
<h2 class="mp-h2" id="mp-tfl-h2"><span id="From_today.27s_featured_list"></span
><span class="mw-headline" id="From_today's_featured_list">From today's feature
d list</span></h2>
<h2 class="mp-h2" id="mp-tfp-h2"><span id="Today.27s_featured_picture"></span><
span class="mw-headline" id="Today's_featured_picture">Today's featured picture
</span></h2>
<h2 class="mp-h2" id="mp-other"><span class="mw-headline" id="Other_areas_of_Wi
kipedia">Other areas of Wikipedia</span></h2>
<h2 class="mp-h2" id="mp-sister"><span id="Wikipedia.27s_sister_projects"></spa
n><span class="mw-headline" id="Wikipedia's_sister_projects">Wikipedia's sister
projects</span></h2>
<h2 class="mp-h2" id="mp-lang"><span class="mw-headline" id="Wikipedia_language
s">Wikipedia languages</span></h2>
<h2>Navigation menu</h2>
<h3 aria-label="" class="vector-menu-heading" id="p-personal-label">
<span>Personal tools</span>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-namespaces-label">
<span>Namespaces</span>
</h3>
<h3 aria-label="Change language variant" class="vector-menu-heading" id="p-vari
ants-label">
<span>Variants</span>
<span class="vector-menu-checkbox-expanded">expanded</span>
<span class="vector-menu-checkbox-collapsed">collapsed</span>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-views-label">
<span>Views</span>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-cactions-label">
<span>More</span>
<span class="vector-menu-checkbox-expanded">expanded</span>
<span class="vector-menu-checkbox-collapsed">collapsed</span>
</h3>
<h3>
<label for="searchInput">Search</label>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-navigation-label">
<span>Navigation</span>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-interaction-label">
<span>Contribute</span>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-tb-label">
<span>Tools</span>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-coll-print_export-label">
<span>Print/export</span>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-wikibase-otherprojects-labe
```

```
l">
<span>In other projects</span>
</h3>
<h3 aria-label="" class="vector-menu-heading" id="p-lang-label">
<span>Languages</span>
</h3>
```

2)MDB’s Top rated 100 movies’ data i.e. Name, IMDB rating, Year

```
In [86]: from bs4 import BeautifulSoup
```

```
In [87]: import requests
```

```
In [88]: import pandas as pd
```

```
In [89]: page=requests.get("https://www.imdb.com/chart/top/")
```

```
In [90]: page
```

```
Out[90]: <Response [200]>
```

```
In [91]: soup=BeautifulSoup(page.text,'html.parser')
```

```
In [92]: print(soup.prettify())
```

...

```
In [93]: wanted_movieswithyear=soup.find_all('td',class_='titleColumn')
```

```
In [94]: wanted_movieswithyear
```

...

```
In [95]: movies=[]
```

In [96]:
```
for movie in wanted_movieswithyear:
    movie=movie.get_text().replace('\n',"")
    movie=movie.strip(" ")
    movies.append(movie)
movies
```

```
 '53.      Apocalypse Now(1979)',
 '54.      Memento(2000)',
 '55.      Raiders of the Lost Ark(1981)',
 '56.      The Great Dictator(1940)',
 '57.      The Lives of Others(2006)',
 '58.      Django Unchained(2012)',
 '59.      Paths of Glory(1957)',
 '60.      Sunset Blvd.(1950)',
 '61.      WALL·E(2008)',
 '62.      Avengers: Infinity War(2018)',
 '63.      Witness for the Prosecution(1957)',
 '64.      The Shining(1980)',
 '65.      Spider-Man: Into the Spider-Verse(2018)',
 '66.      Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb
(1964)',
 '67.      Joker(2019)',
 '68.      Mononoke-hime(1997)',
 '69.      Oldeuboi(2003)',
 '70.      Kimi no na wa.(2016)',
 '71.      The Dark Knight Rises(2012)',
```

In [97]:
```
wanted_ratings=soup.find_all("td",class_='ratingColumn imdbRating')
```

In [98]:
```
wanted_ratings
```

...

In [99]:
```
ratings=[]
for rating in wanted_ratings:
    rating=rating.get_text().replace('\n',"")
    ratings.append(rating)
```

In [100]:
```
ratings
```

Out[100]:
```
['9.2',
 '9.1',
 '9.0',
 '9.0',
 '8.9',
 '8.9',
 '8.9',
 '8.8',
 '8.8',
 '8.8',
 '8.8',
 '8.7',
 '8.7',
 '8.7',
 '8.7',
 '8.6',
 '8.6',
 '8.6',
 '8.6',
```

In [101]:
```
data=pd.DataFrame()
data['Movie with date']=movies
data['Ratings']=ratings
```

In [102]:
```
data.head(100)
```

Out[102]:

	Movie with date	Ratings
0	1. The Shawshank Redemption(1994)	9.2
1	2. The Godfather(1972)	9.1
2	3. The Godfather: Part II(1974)	9.0
3	4. The Dark Knight(2008)	9.0
4	5. 12 Angry Men(1957)	8.9
...	...	...
95	96. Requiem for a Dream(2000)	8.3
96	97. Singin' in the Rain(1952)	8.3
97	98. North by Northwest(1959)	8.3
98	99. Eternal Sunshine of the Spotless Mind...	8.3
99	100. Ladri di biciclette(1948)	8.3

100 rows × 2 columns

3)rating for 100 indian films

In [103]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [104]:
```
page=requests.get("https://www.imdb.com/india/top-rated-indian-movies/")
```

In [105]:
```
page
```

Out[105]:
```
<Response [200]>
```

In [106]:
```
soup=BeautifulSoup(page.text,'html.parser')
```

In [107]:
```
print(soup.prettify())
```

...

In [108]:
```
wanted_indianmoviesanddate=soup.find_all('td',class_='titleColumn')
```

In [109]:
```
wanted_indianmoviesanddate
```

Out[109]:
```
[<td class="titleColumn">
       1.
       <a href="/title/tt10280296/" title="Shoojit Sircar (dir.), Vicky Kausha
l, Banita Sandhu">Sardar Udham</a>
 <span class="secondaryInfo">(2021)</span>
 </td>,
 <td class="titleColumn">
       2.
       <a href="/title/tt0093603/" title="Mani Ratnam (dir.), Kamal Haasan, Sar
anya Ponvannan">Nayakan</a>
 <span class="secondaryInfo">(1987)</span>
 </td>,
 <td class="titleColumn">
       3.
       <a href="/title/tt0367495/" title="Sundar C. (dir.), Kamal Haasan, Madha
van">Anbe Sivam</a>
 <span class="secondaryInfo">(2003)</span>
 </td>,
 <td class="titleColumn">
```

In [110]:
```
movies=[]
```

In [111]:
```
for movie in wanted_indianmoviesanddate:
    movie=movie.get_text().replace('\n',"")
    movie=movie.strip("  ")
    movies.append(movie)
```

Type *Markdown* and LaTeX: α^2

In [112]:
```
movies
```

Out[112]:
```
['1.      Sardar Udham(2021)',
 '2.      Nayakan(1987)',
 '3.      Anbe Sivam(2003)',
 '4.      Pariyerum Perumal(2018)',
 '5.      C/o Kancharapalem(2018)',
 '6.      Manichitrathazhu(1993)',
 '7.      Golmaal(1979)',
 '8.      Kireedam(1989)',
 '9.      Apur Sansar(1959)',
 '10.      Natsamrat(2016)',
 '11.      96(2018)',
 '12.      Thevar Magan(1992)',
 '13.      Kumbalangi Nights(2019)',
 '14.      Black Friday(2004)',
 '15.      Pather Panchali(1955)',
 '16.      Soorarai Pottru(2020)',
 '17.      #Home(2021)',
 '18.      Visaaranai(2015)',
 '19.      3 Idiots(2009)',
```

In [113]:
```
wanted_indianratings=soup.find_all('td',class_="ratingColumn imdbRating")
```

In [114]:
```
wanted_indianratings
```

Out[114]:
```
[<td class="ratingColumn imdbRating">
 <strong title="8.6 based on 20,122 user ratings">8.6</strong>
 </td>,
 <td class="ratingColumn imdbRating">
 <strong title="8.5 based on 18,432 user ratings">8.5</strong>
 </td>,
 <td class="ratingColumn imdbRating">
 <strong title="8.5 based on 18,963 user ratings">8.5</strong>
 </td>,
 <td class="ratingColumn imdbRating">
 <strong title="8.5 based on 12,458 user ratings">8.5</strong>
 </td>,
 <td class="ratingColumn imdbRating">
 <strong title="8.5 based on 6,030 user ratings">8.5</strong>
 </td>,
 <td class="ratingColumn imdbRating">
 <strong title="8.5 based on 9,843 user ratings">8.5</strong>
 </td>,
 <td class="ratingColumn imdbRating">
```

In [115]:
```
ratings=[]
for rating in wanted_indianratings:
    rating=rating.get_text().replace('\n',"")
    ratings.append(rating)
```

In [116]:
```
ratings
```

Out[116]:
```
['8.6',
 '8.5',
 '8.5',
 '8.5',
 '8.5',
 '8.5',
 '8.5',
 '8.5',
 '8.5',
 '8.4',
 '8.4',
 '8.4',
 '8.4',
 '8.4',
 '8.4',
 '8.4',
 '8.4',
 '8.4',
 '8.4',
```

In [117]:
```
data2=pd.DataFrame()
data2['Movie with date']=movies
data2['Ratings']=ratings
data2.head(100)
```

Out[117]:

	Movie with date	Ratings
0	1. Sardar Udham(2021)	8.6
1	2. Nayakan(1987)	8.5
2	3. Anbe Sivam(2003)	8.5
3	4. Pariyerum Perumal(2018)	8.5
4	5. C/o Kancharapalem(2018)	8.5
...	...	...
95	96. Roja(1992)	8.1
96	97. Dil Chahta Hai(2001)	8.1
97	98. Rang De Basanti(2006)	8.1
98	99. OMG: Oh My God!(2012)	8.1
99	100. Uri: The Surgical Strike(2019)	8.1

100 rows × 2 columns

4)a)MENS 0DI COUNTRY RANKING

In [118]:
```
from bs4 import BeautifulSoup
```

In [119]:
```
import requests
```

In [120]:
```
import pandas as pd
```

In [121]:
```
page=requests.get("https://www.icc-cricket.com/rankings/mens/team-rankings/odi")
```

In [122]:
```
page
```

Out[122]:
```
<Response [200]>
```

In [123]:
```
soup=BeautifulSoup(page.text,'html.parser')
```

In [124]:
```
print(soup.prettify())
```

...

In [125]:
```
table=soup.find("table",{"class":"table"})
```

In [126]:
```
table
```

...

```
In [127]: trs=table.find_all("tr")
```

```
In [128]: rows=[]
          columns=['POS','TEAM','MATCHES','POINTS','RATING']
```

```
In [129]: for tr in trs:
              tds=tr.find_all('td')
              row=[td.text.replace('\n','').strip()for td in tds]
              rows.append(row)
```

```
In [130]: df=pd.DataFrame(rows,columns=columns)
```

```
In [131]: print(df.head(11))
```

```
     POS            TEAM MATCHES  POINTS RATING
0   None            None    None    None   None
1      1   New ZealandNZ      17   2,054    121
2      2      EnglandENG      32   3,793    119
3      3    AustraliaAUS      28   3,244    116
4      4        IndiaIND      32   3,624    113
5      5  South AfricaSA      25   2,459     98
6      6     PakistanPAK      27   2,524     93
7      7   BangladeshBAN      30   2,740     91
8      8   West IndiesWI      30   2,523     84
9      9     Sri LankaSL      32   2,657     83
10    10  AfghanistanAFG      17   1,054     62
```

4)b)mens batting ranking

```
In [132]: from bs4 import BeautifulSoup
          import requests
          import pandas as pd
```

```
In [133]: page=requests.get("https://www.icc-cricket.com/rankings/mens/player-rankings/odi"
```

```
In [134]: page
```

```
Out[134]: <Response [200]>
```

```
In [135]: soup=BeautifulSoup(page.text,'html.parser')
          print(soup.prettify)
```

...

```
In [136]: table=soup.find("table",{"class":"table"})
```

In [137]:
```
table
```

Out[137]:
```
<table class="table">
<thead>
<tr class="table-head">
<th class="table-head__cell u-text-right">Pos</th>
<th class="table-head__cell">Player</th>
<th class="table-head__cell">Team</th>
<th class="table-head__cell u-text-right">Rating</th>
</tr>
</thead>
<tbody>
<tr class="table-body">
<td class="table-body__cell table-body__cell--position u-text-right">
                                  2
        <span class="ranking-pos no-change"></span>
</td>
<td class="table-body__cell name">
<a href="/rankings/mens/player-rankings/164">Virat Kohli</a>
</td>
<td class="table-body__cell nationality-logo">
```

In [138]:
```
trs=table.find_all("tr")
rows=[]
columns=['POS','PLAYER','TEAM','RATING']
```

In [139]:
```
for tr in trs:
    tds=tr.find_all("td")
    row=[td.text.replace("\n","").strip()for td in tds]
    rows.append(row)
```

In [140]:
```
df=pd.DataFrame(rows,columns=columns)
```

In [141]:
```
df.head(10)
```

Out[141]:

	POS	PLAYER	TEAM	RATING
0	None	None	None	None
1	2	Virat Kohli	IND	844
2	3	Rohit Sharma	IND	813
3	4	Ross Taylor	NZ	801
4	5	Aaron Finch	AUS	779
5	6	Jonny Bairstow	ENG	775
6	7	David Warner	AUS	762
7	8	Shai Hope	WI	758
8	9	Kane Williamson	NZ	754
9	10	Quinton de Kock	SA	747

4)c)men bowling rank

```
In [142]: from bs4 import BeautifulSoup
          import requests
          import pandas as pd
```

```
In [143]: page=requests.get("https://www.icc-cricket.com/rankings/mens/player-rankings/odi/
          page

Out[143]: <Response [200]>
```

```
In [144]: soup=BeautifulSoup(page.text,'html.parser')
          print(soup.prettify)
```

...

```
In [145]: table=soup.find("table",{"class":"table rankings-table"})
          table

Out[145]: <table class="table rankings-table">
          <thead>
          <tr class="table-head">
          <th class="table-head__cell u-text-right">Pos</th>
          <th class="table-head__cell">Player</th>
          <th class="table-head__cell">Team</th>
          <th class="table-head__cell table-head__cell--rating">Rating</th>
          <th class="table-head__cell u-text-right u-hide-phablet">Career Best Rating</th
          >
          </tr>
          </thead>
          <tr class="rankings-block__banner">
          <td class="rankings-block__position">1</td>
          <td class="rankings-block__top-player-container">
          <div class="u-flex-center u-text-left">
          <div class="rankings-block__player-image-container rankings-block__player-image
          -container--large">
          <a href="/rankings/mens/player-rankings/969">
          <img alt="Player Image" class="rankings-block__player-image image-missing" data
```

```
In [146]: trs=table.find_all("tr")
```

```
In [147]: rows=[]
          columns=['POS','PLAYER','TEAM','RATING','BEST']
```

```
In [148]: for tr in trs:
              tds=tr.find_all("td")
              row=[td.text.replace('\n','').strip()for td in tds]
              rows.append(row)
```

```
In [149]: df=pd.DataFrame(rows,columns=columns)
In [150]: df
```

Out[150]:

	POS	PLAYER	TEAM	RATING	BEST
0	None	None	None	None	None
1	1	Trent Boult	NZ	737	770 v West Indies, 22/06/2019
2	2	Josh Hazlewood	AUS	709	733 v England, 26/01/2018
3	3	Mujeeb Ur Rahman	AFG	708	712 v Ireland, 24/01/2021
4	4	Chris Woakes	ENG	700	711 v Sri Lanka, 04/07/2021
...	...	...	...	...	...
96	96	Oshane Thomas	WI	355	419 v Australia, 06/06/2019
97	97	Kyle Jamieson	NZ	352	352 v Bangladesh, 26/03/2021
98	98	Charles Amini	PNG	351	351 v Nepal, 10/09/2021
99		Michael Leask	SCO	351	362 v Papua New Guinea, 17/08/2019
100	100	Sharafuddin Ashraf	AFG	347	354 v West Indies, 25/03/2018

101 rows × 5 columns

5)a)womens team ranking

```
In [151]: from bs4 import BeautifulSoup
          import requests
          import pandas as pd
```

```
In [152]: page=requests.get("https://www.icc-cricket.com/rankings/womens/team-rankings/odi"
```

```
In [153]: page
```

```
Out[153]: <Response [200]>
```

```
In [154]: soup=BeautifulSoup(page.text,'html.parser')
```

```
In [155]: print(soup.prettify)
```

```
...
```

```
In [156]: tablew=soup.find("table",{"class":"table"})
In [157]: tablew
```

```
Out[157]: <table class="table">
<thead>
<tr class="table-head">
<th class="table-head__cell u-text-right">Pos</th>
<th class="table-head__cell rankings-table__team">
<span class="u-hide-mobile">Team</span>
<span class="u-show-mobile">T</span>
</th>
<th class="table-head__cell u-center-text">
<span class="u-hide-mobile">Matches</span>
<span class="u-show-mobile">M</span>
</th>
<th class="table-head__cell u-center-text">
<span class="u-hide-mobile">Points</span>
<span class="u-show-mobile">P</span>
</th>
<th class="table-head__cell u-text-right rating">
<span class="u-hide-mobile">Rating</span>
<span class="u-show-mobile">R</span>
</th>
```

```
In [158]: trs=tablew.find_all("tr")
```

```
In [159]: rows=[]
```

```
In [160]: columns=['POS','TEAM','MATCHES','POINT','RATING']
```

```
In [161]: for tr in trs:
              tds=tr.find_all("td")
              row=[td.text.replace('\n','').strip()for td in tds]
              rows.append(row)
```

```
In [162]: df=pd.DataFrame(rows,columns=columns)
```

```
In [163]: df.head(11)
```

```
Out[163]:
      POS           TEAM  MATCHES  POINT  RATING
 0   None           None     None   None    None
 1      1     AustraliaAUS       21  3,379     161
 2      2       EnglandENG       25  2,983     119
 3      3  South AfricaSA       29  3,390     117
 4      4        IndiaIND       26  2,934     113
 5      5   New ZealandNZ       26  2,392      92
 6      6   West IndiesWI       22  1,872      85
 7      7     PakistanPAK       20  1,496      75
 8      8   BangladeshBAN        5    306      61
 9      9     Sri LankaSL       11    519      47
10     10       IrelandIRE        2     25      13
```

5)b)women top 10 odi player(batting)

In [164]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [165]:
```
page=requests.get("https://www.icc-cricket.com/rankings/womens/player-rankings/od
```

In [166]:
```
page
```

Out[166]:
```
<Response [200]>
```

In [167]:
```
soup=BeautifulSoup(page.text,'html.parser')
```

In [168]:
```
print(soup.prettify)
```

...

In [169]:
```
tablew1=soup.find("table",{"class":"table"})
```

Type *Markdown* and LaTeX: α^2

In [170]:
```
tablew1
```

Out[170]:
```
<table class="table">
<thead>
<tr class="table-head">
<th class="table-head__cell u-text-right">Pos</th>
<th class="table-head__cell">Player</th>
<th class="table-head__cell">Team</th>
<th class="table-head__cell u-text-right">Rating</th>
</tr>
</thead>
<tbody>
<tr class="table-body">
<td class="table-body__cell table-body__cell--position u-text-right">
                                    2
        <span class="ranking-pos no-change"></span>
</td>
<td class="table-body__cell name">
<a href="/rankings/womens/player-rankings/466">Alyssa Healy</a>
</td>
<td class="table-body__cell nationality-logo">
```

In [171]:
```
trs=tablew1.find_all("tr")
```

In [172]:
```
rows=[]
columns=['POS','PLAYER','TEAM','RATING']
```

In [173]:
```
for tr in trs:
    tds=tr.find_all("td")
    row=[td.text.replace('\n','').strip()for td in tds]
    rows.append(row)
```

In [174]:
```
df=pd.DataFrame(rows,columns=columns)
```

In [175]:
```
df.head(11)
```

Out[175]:

	POS	PLAYER	TEAM	RATING
0	None	None	None	None
1	2	Alyssa Healy	AUS	750
2	3	Mithali Raj	IND	738
3	4	Tammy Beaumont	ENG	728
4	5	Amy Satterthwaite	NZ	717
5	6	Smriti Mandhana	IND	710
6	7	Meg Lanning	AUS	699
7	8	Beth Mooney	AUS	690
8	9	Heather Knight	ENG	674
9	10	Laura Wolvaardt	SA	672

5)c)womens bowling

In [176]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [177]:
```
page=requests.get("https://www.icc-cricket.com/rankings/womens/player-rankings/od
```

In [178]:
```
page
```

Out[178]:
```
<Response [200]>
```

In [179]:
```
soup=BeautifulSoup(page.text,'html.parser')
print(soup.prettify())
```

...

In [180]:
```
tablew2=soup.find("table",{"class":"table"})
tablew2
```

Out[180]:
```
<table class="table rankings-table">
<thead>
<tr class="table-head">
<th class="table-head__cell u-text-right">Pos</th>
<th class="table-head__cell">Player</th>
<th class="table-head__cell">Team</th>
<th class="table-head__cell table-head__cell--rating">Rating</th>
<th class="table-head__cell u-text-right u-hide-phablet">Career Best Rating</th
>
</tr>
</thead>
<tr class="rankings-block__banner">
<td class="rankings-block__position">1</td>
<td class="rankings-block__top-player-container">
<div class="u-flex-center u-text-left">
<div class="rankings-block__player-image-container rankings-block__player-image
-container--large">
<a href="/rankings/womens/player-rankings/468">
<img alt="Player Image" class="rankings-block__player-image image-missing" data
```

In [181]:
```
trs=tablew2.find_all("tr")
```

In [182]:
```
rows=[]
columns=['POS','PLAYER','TEAM','RATING','BEST']
```

In [183]:
```
for tr in trs:
    tds=tr.find_all("td")
    row=[td.text.replace('\n','').strip()for td in tds]
    rows.append(row)
```

In [184]:
```
df=pd.DataFrame(rows,columns=columns)
```

In [185]:
```
print(df.head(11))
```

```
     POS              PLAYER  TEAM RATING                           BEST
0   None                None  None   None                           None
1      1       Jess Jonassen   AUS    760  808 v New Zealand, 10/04/2021
2      2      Jhulan Goswami   IND    727      796 v England, 28/02/2007
3      3        Megan Schutt   AUS    717  766 v West Indies, 11/09/2019
4      4      Marizanne Kapp    SA    715  749 v West Indies, 10/09/2021
5      5  Sophie Ecclestone   ENG    701  701 v New Zealand, 26/09/2021
6      6      Shabnim Ismail    SA    688        743 v India, 07/03/2021
7      7    Katherine Brunt   ENG    666        796 v India, 03/02/2013
8      8      Ayabonga Khaka    SA    643  662 v West Indies, 10/09/2021
9      9      Anya Shrubsole   ENG    598     655 v Pakistan, 12/12/2019
10    10          Kate Cross   ENG    589  589 v New Zealand, 26/09/2021
```

6)mobile data from Amazon

In [53]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [54]:
```
page=requests.get("https://www.amazon.in/s?k=Mobile+phones+under+20000&ref=nb_sb_
In [55]: page
```

```
Out[55]: <Response [200]>
```

```
In [4]: soup=BeautifulSoup(page.text,'html.parser')
```

```
In [5]: print(soup.prettify())
```

...

```
In [56]: names=[]
```

```
In [57]: for i in soup.find_all('span',class_="a-size-medium a-color-base a-text-normal"):
             names.append(i.text.split("|"))
```

```
In [78]: titles=names[0:14]
```

```
In [79]: titles
```

```
Out[79]: [['Redmi 9 (Sky Blue, 4GB RAM, 64GB Storage) ',
           ' 2.3GHz Mediatek Helio G35 Octa core Processor'],
          ['Redmi 9A (Nature Green, 2GB RAM, 32GB Storage) ',
           ' 2GHz Octa-core Helio G25 Processor ',
           ' 5000 mAh Battery'],
          ['OPPO A31 (Fantasy White, 6GB RAM, 128GB Storage) with No Cost EMI/Additional
         Exchange Offers'],
          ['Samsung Galaxy M31 (Ocean Blue, 8GB RAM, 128GB Storage) 6 Months Free Screen
         Replacement for Prime'],
          ['OPPO A74 5G (Fantastic Purple,6GB RAM,128GB Storage) - 5G Android Smartphone
         ',
           ' 5000 mAh Battery ',
           ' 18W Fast Charge ',
           ' 90Hz LCD Display'],
          ['Samsung Galaxy M12 (Blue,4GB RAM, 64GB Storage) 6000 mAh with 8nm Processor
         ',
           ' True 48 MP Quad Camera ',
           ' 90Hz Refresh Rate'],
          ['OPPO A31 (Mystery Black, 6GB RAM, 128GB Storage) with No Cost EMI/Additional
         Exchange Offers'],
          ['Samsung Galaxy M12 (Black,4GB RAM, 64GB Storage) 6000 mAh with 8nm Processor
         ',
           ' True 48 MP Quad Camera ',
           ' 90Hz Refresh Rate'],
          ['Samsung Galaxy M32 5G (Sky Blue, 6GB RAM, 128GB Storage)'],
          ['Samsung Galaxy M32 5G (Slate Black, 6GB RAM, 128GB Storage)'],
          ['Redmi 9A (Nature Green, 3GB Ram, 32GB Storage) ',
           ' 2GHz Octa-core Helio G25 Processor'],
          ['Samsung Galaxy M32 5G (Sky Blue, 8GB RAM, 128GB Storage)'],
          ['Redmi 9A (Midnight Black, 2GB RAM, 32GB Storage) ',
           ' 2GHz Octa-Core Helio G25 Processor'],
          ['Samsung Galaxy M12 (White,4GB RAM, 64GB Storage) 6000 mAh with 8nm Processor
         ',
           ' True 48 MP Quad Camera ',
           ' 90Hz Refresh Rate']]
```

```
In [60]: price=[]
```

```
In [61]: for i in soup.find_all('span',class_="a-price-whole"):
             price.append(i.text)
```

In [80]:
```
rate=price[0:14]
```

In [81]:
```
rate
```

Out[81]:
```
['8,499',
 '6,999',
 '11,490',
 '15,999',
 '15,990',
 '9,499',
 '11,490',
 '9,499',
 '16,999',
 '16,999',
 '7,799',
 '18,999',
 '6,999',
 '9,499']
```

In [64]:
```
imageurls=[]
```

In [65]:
```
for i in soup.find_all('img',class_="s-image"):
    imageurls.append(i['src'])
```

In [66]:
```
image=imageurls[0:14]
```

In [73]:
```
image
```

Out[73]:
```
['https://m.media-amazon.com/images/I/71A9Vo1BatL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71sxlhYhKWL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/61CnyJ-IbML._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71-Su4Wr0HL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71geVdy6-OS._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71r69Y7BSeL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71KCwNV6MuL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/7162Y5fPdkL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71os5DRhuSL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71QT7dSK4BL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71sxlhYhKWL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71os5DRhuSL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71sxlhYhKWL._AC_UY218_.jpg',
 'https://m.media-amazon.com/images/I/71Y8rH2cJiL._AC_UY218_.jpg']
```

In [74]:
```
ratings=[]
```

In [75]:
```
for i in soup.find_all('i',class_='a-icon a-icon-star-small a-star-small-4 aok-al
    ratings.append(i.text)
```

```
In [82]: rating=ratings[0:14]
         rating
```

```
Out[82]: ['4.2 out of 5 stars',
          '4.2 out of 5 stars',
          '4.2 out of 5 stars',
          '4.2 out of 5 stars',
          '4.2 out of 5 stars',
          '4.1 out of 5 stars',
          '4.2 out of 5 stars',
          '4.1 out of 5 stars',
          '3.8 out of 5 stars',
          '3.8 out of 5 stars',
          '4.2 out of 5 stars',
          '3.8 out of 5 stars',
          '4.2 out of 5 stars',
          '4.1 out of 5 stars']
```

```
In [83]: df=pd.DataFrame({'NAMES':titles,"PRICE":rate,"IMAGE":image,"RATING":rating})
```

```
In [84]: df
```

Out[84]:

	NAMES	PRICE	IMAGE	RATING
0	[Redmi 9 (Sky Blue, 4GB RAM, 64GB Storage) , ...	8,499	https://m.media-amazon.com/images/I/71A9Vo1Bat...	4.2 out of 5 stars
1	[Redmi 9A (Nature Green, 2GB RAM, 32GB Storage...	6,999	https://m.media-amazon.com/images/I/71sxlhYhKW...	4.2 out of 5 stars
2	[OPPO A31 (Fantasy White, 6GB RAM, 128GB Stora...	11,490	https://m.media-amazon.com/images/I/61CnyJ-IbM...	4.2 out of 5 stars
3	[Samsung Galaxy M31 (Ocean Blue, 8GB RAM, 128G...	15,999	https://m.media-amazon.com/images/I/71-Su4Wr0H...	4.2 out of 5 stars
4	[OPPO A74 5G (Fantastic Purple,6GB RAM,128GB S...	15,990	https://m.media-amazon.com/images/I/71geVdy6-O...	4.2 out of 5 stars
5	[Samsung Galaxy M12 (Blue,4GB RAM, 64GB Storag...	9,499	https://m.media-amazon.com/images/I/71r69Y7BSe...	4.1 out of 5 stars
6	[OPPO A31 (Mystery Black, 6GB RAM, 128GB Stora...	11,490	https://m.media-amazon.com/images/I/71KCwNV6Mu...	4.2 out of 5 stars
7	[Samsung Galaxy M12 (Black,4GB RAM, 64GB Stora...	9,499	https://m.media-amazon.com/images/I/7162Y5fPdk...	4.1 out of 5 stars
8	[Samsung Galaxy M32 5G (Sky Blue, 6GB RAM, 128...	16,999	https://m.media-amazon.com/images/I/71os5DRhuS...	3.8 out of 5 stars
9	[Samsung Galaxy M32 5G (Slate Black, 6GB RAM, ...	16,999	https://m.media-amazon.com/images/I/71QT7dSK4B...	3.8 out of 5 stars
10	[Redmi 9A (Nature Green, 3GB Ram, 32GB Storage...	7,799	https://m.media-amazon.com/images/I/71sxlhYhKW...	4.2 out of 5 stars
11	[Samsung Galaxy M32 5G (Sky Blue, 8GB RAM, 128...	18,999	https://m.media-amazon.com/images/I/71os5DRhuS...	3.8 out of 5 stars
12	[Redmi 9A (Midnight Black, 2GB RAM, 32GB Stora...	6,999	https://m.media-amazon.com/images/I/71sxlhYhKW...	4.2 out of 5 stars
13	[Samsung Galaxy M12 (White,4GB RAM, 64GB Stora...	9,499	https://m.media-amazon.com/images/I/71Y8rH2cJi...	4.1 out of 5 stars

7)home data from nobroker site

In [86]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [87]:
```
page=requests.get("https://www.nobroker.in/property/sale/bangalore/Electronic Cit
```

In [88]:
```
page
```

Out[88]:
```
<Response [200]>
```

In [89]:
```
soup=BeautifulSoup(page.text,"html.parser")
```

In [90]:
```
print(soup.prettify())
```

...

In [91]:
```
housenames=[]
```

In [92]:
```
for i in soup.find_all('a',class_="nb__3CnI6"):
    housenames.append(i.text)
```

In [93]:
```
housenames
```

Out[93]:
```
['4 BHK In Independent House  For Sale  In Hebbagodi ',
 '4 BHK Apartment  For Sale  In Nisarga Residency In Electronic City Phase Ii
',
 '4 BHK Flat  For Sale  In Sobha Silicon Oasis  In Hosa Road ',
 '4 BHK For Sale  In Daadys Garden In Electronic City ',
 '4 BHK Flat  For Sale  In , Electronic City ',
 '4 BHK Flat  For Sale  In Hosa Road, Parappana Agrahara ',
 '4 BHK In Independent House  For Sale  In Electronic City ',
 '4 BHK In Independent House  For Sale  In Electronic City ',
 '4 BHK Apartment  For Sale  In Gopalan Gardenia In Electronic City ',
 '4 BHK For Sale  In Gpr Royale In Gpr Royale ']
```

In [94]:
```
locations=[]
```

In [95]:
```
for i in soup.find_all('div',class_="nb__2CMjv"):
    locations.append(i.text)
```

In [96]:
```
locations
```

Out[96]:
```
['Independent House, Bangalore - Hosur Road, Near  National Public School',
 'Nisarga ResidencyÂ\xa0 Near Thali Resturant, Ananth Nagar, Electronic City Ph
ase II, Bangalore, Karnataka, INDIA.',
 'Sobha Silicon Oasis Naganathapura, Rayasandra Bengaluru, Karnataka 560100 Ind
ia',
 'Daadys GardenÂ\xa0 Kammasandra Rd, Kammasandra, Electronic City, Bengaluru, K
arnataka 560100, India',
 'Standalone Building, 16th Cross Road Neeladri Nagar, near by brand factory',
 'Standalone Building, 11th cross.anjanadri lay out',
 'Independent House, surya nagar face 1',
 'Independent House, Hosur Rd,Near Infosys Limited',
 'Gopalan GardeniaÂ\xa0 Gopalan gardenia, Veerasandra Main Rd, Veer Sandra, Ele
ctronic City, Bengaluru, Karnataka 560100, India',
 '6th Cross']
```

In [97]:
```
area=[]
```

In [98]:

```python
for i in soup.find_all('div',class_="nb__3oNyC"):
    area.append(i.text)
```

In [99]:

```python
area
```

Out[99]:

```
['1,800 sqft',
 '2,000 sqft',
 '1,879 sqft',
 '2,600 sqft',
 '2,000 sqft',
 '3,000 sqft',
 '3,000 sqft',
 '1,200 sqft',
 '2,650 sqft',
 '3,100 sqft']
```

In [101]:

```python
df=pd.DataFrame({"HOUSENAME":housenames,"LOCATION":locations,"AREA":area})
```

In [102]:

```python
df
```

Out[102]:

	HOUSENAME	LOCATION	AREA
0	4 BHK In Independent House For Sale In Hebba...	Independent House, Bangalore - Hosur Road, Nea...	1,800 sqft
1	4 BHK Apartment For Sale In Nisarga Residenc...	Nisarga ResidencyÂ Near Thali Resturant, Anan...	2,000 sqft
2	4 BHK Flat For Sale In Sobha Silicon Oasis ...	Sobha Silicon Oasis Naganathapura, Rayasandra ...	1,879 sqft
3	4 BHK For Sale In Daadys Garden In Electronic...	Daadys GardenÂ Kammasandra Rd, Kammasandra, E...	2,600 sqft
4	4 BHK Flat For Sale In , Electronic City	Standalone Building, 16th Cross Road Neeladri ...	2,000 sqft
5	4 BHK Flat For Sale In Hosa Road, Parappana ...	Standalone Building, 11th cross.anjanadri lay out	3,000 sqft
6	4 BHK In Independent House For Sale In Elect...	Independent House, surya nagar face 1	3,000 sqft
7	4 BHK In Independent House For Sale In Elect...	Independent House, Hosur Rd,Near Infosys Limited	1,200 sqft
8	4 BHK Apartment For Sale In Gopalan Gardenia...	Gopalan GardeniaÂ Gopalan gardenia, Veerasand...	2,650 sqft
9	4 BHK For Sale In Gpr Royale In Gpr Royale	6th Cross	3,100 sqft

8)restaurent informations

In [224]:

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

In [225]:

```python
page=requests.get("https://www.dineout.co.in/delhi-restaurants/buffet-special")
page
```

Out[225]:

```
<Response [200]>
```

```
In [226]: soup=BeautifulSoup(page.text,'html.parser')
```

```
In [227]: print(soup.prettify())
```

```
...
```

```
In [228]: titles=[]
          for i in soup.find_all('div',class_="restnt-info cursor"):
              titles.append(i.text.split(',')[0])
```

```
In [229]: titles
```

```
Out[229]: ['Castle BarbequeConnaught Place',
           'Jungle Jamboree3CS Mall',
           'Castle BarbequePacific Mall',
           'The Barbeque CompanyGardens Galleria',
           'Cafe KnoshThe Leela Ambience Convention Hotel',
           'India GrillHilton Garden Inn',
           'Delhi BarbequeTaurus Sarovar Portico',
           'The Monarch - Bar Be Que VillageIndirapuram Habitat Centre',
           'World CafeVibe by The Lalit Traveller',
           'Indian Grill RoomSuncity Business Tower',
           'Mad 4 Bar B QueSector 29',
           'Barbeque 29NIT',
           'GlasshouseDoubleTree By Hilton Gurugram Baani Square']
```

```
In [230]: cuisine=[]
```

```
In [231]: for i in soup.find_all('span',class_="double-line-ellipsis"):
              cuisine.append(i.text.split('|')[1])
```

```
In [232]: cuisine
```

```
Out[232]: [' Chinese, North Indian',
           ' North Indian, Barbecue, Italian, Asian',
           ' North Indian, Chinese',
           ' Barbecue, Chinese, Mughlai, North Indian',
           ' Multi-Cuisine, North Indian, Italian, Continental, Mediterranean',
           ' North Indian, Italian, Oriental ',
           ' Barbecue, North Indian',
           ' North Indian, Chinese, Fast Food',
           ' North Indian, Chinese, Continental',
           ' North Indian, Mughlai, Barbecue',
           ' North Indian, Mughlai',
           ' Barbecue, Chinese, North Indian',
           ' Multi-Cuisine, Asian, European, Italian, North Indian']
```

```
In [233]: locations=[]
```

```
In [234]: for i in soup.find_all('div',class_="restnt-info cursor"):
              locations.append(i.text.split(',')[1])
```

In [235]:
```
locations
```

Out[235]:
```
[' Central Delhi',
 'Lajpat Nagar - 3',
 'Tagore Garden',
 'Sector 38A',
 'Shahdara',
 'Saket',
 'Mahipalpur',
 'Indirapuram',
 'Sector 35',
 'Golf Course Road',
 ' Faridabad',
 ' Faridabad',
 'Sector 50']
```

In [236]:
```
imageurls=[]
```

In [237]:
```
for i in soup.find_all('img',class_="no-img"):
    imageurls.append(i['data-src'])
```

In [238]:
```
imageurls
```

Out[238]:
```
['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/8/k/b/p86792-1606
2953735fbe1f4d3fb7e.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/a/k/p59633-1604
6474755fa4fa33c0e92.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/j/o/p38113-1595
9192065f1fcb666130c.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/7/q/d/p79307-1605
1787075fad15532bd7c.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/j/v/p406-163401
663361651d79326d0.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/2/v/t/p2687-14824
77169585cce712b90f.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/v/f/p52501-1600
6856545f68865616659.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/n/o/p34822-1559
9107305cfa594a13c24.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/p/y/p12366-1466
935020576fa6ecdc359.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/y/y/p549-151437
67525a438e30b3e19.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/j/e/p43488-1529
5778165b2b8158ceeef.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/w/r/p58842-1562
4171585d209806d9143.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/9/m/f/p9875-16057
921085fb6716cc44f8.jpg?tr=tr:n-medium']
```

In [239]:
```
len(imageurls)
```

Out[239]:
```
13
```

In [240]:
```
rating=[]
```

```
In [241]: for i in soup.find_all("div",class_="restnt-rating rating-3"):
              rating.append(i.text)
          for i in soup.find_all("div",class_="restnt-rating rating-4"):
              rating.append(i.text)
```

```
In [242]: rating
```

```
Out[242]: ['3.4',
           '3.9',
           '4',
           '4.1',
           '4.3',
           '3.9',
           '3.7',
           '3.9',
           '4.3',
           '4.3',
           '3.8',
           '4.3',
           '4']
```

```
In [243]: df=pd.DataFrame({'TITLE':titles,'CUISINE':cuisine,'LOCATION':locations,'IMAGE':im
```

In [244]:
```
df
```

Out[244]:

	TITLE	CUISINE	LOCATION	IMAGE	RATING
0	Castle BarbequeConnaught Place	Chinese, North Indian	Central Delhi	https://im1.dineout.co.in/images/uploads/resta...	3.4
1	Jungle Jamboree3CS Mall	North Indian, Barbecue, Italian, Asian	Lajpat Nagar - 3	https://im1.dineout.co.in/images/uploads/resta...	3.9
2	Castle BarbequePacific Mall	North Indian, Chinese	Tagore Garden	https://im1.dineout.co.in/images/uploads/resta...	4
3	The Barbeque CompanyGardens Galleria	Barbecue, Chinese, Mughlai, North Indian	Sector 38A	https://im1.dineout.co.in/images/uploads/resta...	4.1
4	Cafe KnoshThe Leela Ambience Convention Hotel	Multi-Cuisine, North Indian, Italian, Contine...	Shahdara	https://im1.dineout.co.in/images/uploads/resta...	4.3
5	India GrillHilton Garden Inn	North Indian, Italian, Oriental	Saket	https://im1.dineout.co.in/images/uploads/resta...	3.9
6	Delhi BarbequeTaurus Sarovar Portico	Barbecue, North Indian	Mahipalpur	https://im1.dineout.co.in/images/uploads/resta...	3.7
7	The Monarch - Bar Be Que VillageIndirapuram Ha...	North Indian, Chinese, Fast Food	Indirapuram	https://im1.dineout.co.in/images/uploads/resta...	3.9
8	World CafeVibe by The Lalit Traveller	North Indian, Chinese, Continental	Sector 35	https://im1.dineout.co.in/images/uploads/resta...	4.3
9	Indian Grill RoomSuncity Business Tower	North Indian, Mughlai, Barbecue	Golf Course Road	https://im1.dineout.co.in/images/uploads/resta...	4.3
10	Mad 4 Bar B QueSector 29	North Indian, Mughlai	Faridabad	https://im1.dineout.co.in/images/uploads/resta...	3.8
11	Barbeque 29NIT	Barbecue, Chinese, North Indian	Faridabad	https://im1.dineout.co.in/images/uploads/resta...	4.3
12	GlasshouseDoubleTree By Hilton Gurugram Baani ...	Multi-Cuisine, Asian, European, Italian, Nort...	Sector 50	https://im1.dineout.co.in/images/uploads/resta...	4

9)weather details of last 24 hour

In [103]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [104]:
```
page=requests.get("https://en.tutiempo.net/delhi.html?data=last-24-hours")
```

```
In [105]: page

Out[105]: <Response [200]>
```

```
In [106]: soup=BeautifulSoup(page.text,'html.parser')
```

```
In [107]: print(soup.prettify)
```

...

```
In [110]: hour=[]
```

```
In [111]: for i in soup.find_all("td",class_="t Temp")[0:24]:
              if i.previous_sibling.previous_sibling is not None:
                  hour.append(i.previous_sibling.previous_sibling.text)
              else:
                  hour.append("")
```

```
In [112]: hour

Out[112]: ['15:00',
           '14:30',
           '14:00',
           '13:30',
           '13:00',
           '12:30',
           '12:00',
           '11:30',
           '11:00',
           '10:30',
           '10:00',
           '09:30',
           '09:00',
           '08:30',
           '08:00',
           '07:30',
           '06:00',
           '05:30',
           '05:00',
           '04:30',
           '04:00',
           '03:30',
           '03:00',
           '02:30']
```

```
In [139]: temp=[]
```

```
In [140]: for i in soup.find_all("td",class_="t Temp")[0:24]:
              temp.append(i.text)
```

In [141]:
```
temp
```

Out[141]:
```
['28°C',
 '28°C',
 '28°C',
 '28°C',
 '27°C',
 '27°C',
 '26°C',
 '25°C',
 '23°C',
 '23°C',
 '22°C',
 '22°C',
 '21°C',
 '20°C',
 '19°C',
 '19°C',
 '19°C',
 '19°C',
 '19°C',
 '19°C',
 '19°C',
 '19°C',
 '19°C',
 '19°C']
```

In [142]:
```
wind=[]
```

In [143]:
```
for i in soup.find_all("td",class_="wind")[0:24]:
    wind.append(i.text)
```

In [144]:
```
wind
```

Out[144]:
```
['9 km/h',
 '7 km/h',
 '9 km/h',
 '11 km/h',
 '11 km/h',
 '11 km/h',
 '11 km/h',
 '9 km/h',
 '7 km/h',
 '7 km/h',
 '9 km/h',
 '9 km/h',
 '7 km/h',
 'Calm',
 'Calm',
 '7 km/h',
 '7 km/h',
 '7 km/h',
 'Calm',
 '6 km/h',
 '6 km/h',
 '7 km/h',
 '6 km/h',
 'Calm']
```

In [145]:
```
weather=[]
```

```
In [146]: for i in soup.find_all("span",class_="thhip ico i0530 u303")[0:24]:
              weather.append(i.text)
```

```
In [147]: weather
```

```
Out[147]: ['Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist',
           'Mist']
```

```
In [148]: humidity=[]
```

```
In [149]: for i in soup.find_all("td",class_="hr")[0:24]:
              humidity.append(i.text)
```

```
In [150]: humidity
```

```
Out[150]: ['45%',
           '45%',
           '48%',
           '48%',
           '51%',
           '51%',
           '54%',
           '54%',
           '53%',
           '53%',
           '57%',
           '57%',
           '60%',
           '64%',
           '68%',
           '68%',
           '68%',
           '73%',
           '78%',
           '78%',
           '78%',
           '78%',
           '78%',
           '78%']
```

```
In [151]: pressure=[]
```

```
In [152]: for i in soup.find_all("td",class_="prob")[0:24]:
              pressure.append(i.text)
```

In [153]:

```
pressure
```

Out[153]:

```
['1013 hPa',
 '1014 hPa',
 '1014 hPa',
 '1014 hPa',
 '1015 hPa',
 '1015 hPa',
 '1016 hPa',
 '1017 hPa',
 '1017 hPa',
 '1017 hPa',
 '1017 hPa',
 '1017 hPa',
 '1017 hPa',
 '1017 hPa',
 '1016 hPa',
 '1016 hPa',
 '1016 hPa',
 '1015 hPa',
 '1015 hPa',
 '1015 hPa',
 '1015 hPa',
 '1014 hPa',
 '1014 hPa',
 '1014 hPa']
```

In [154]:

```
df=pd.DataFrame({"HOUR":hour,"TEMP":temp,"WIND":wind,"HUMIDITY":humidity,"PRESSUR
In [155]: df
```

```
Out[155]:
      HOUR  TEMP     WIND  HUMIDITY  PRESSURE
 0   15:00  28°C   9 km/h       45%  1013 hPa
 1   14:30  28°C   7 km/h       45%  1014 hPa
 2   14:00  28°C   9 km/h       48%  1014 hPa
 3   13:30  28°C  11 km/h       48%  1014 hPa
 4   13:00  27°C  11 km/h       51%  1015 hPa
 5   12:30  27°C  11 km/h       51%  1015 hPa
 6   12:00  26°C  11 km/h       54%  1016 hPa
 7   11:30  25°C   9 km/h       54%  1017 hPa
 8   11:00  23°C   7 km/h       53%  1017 hPa
 9   10:30  23°C   7 km/h       53%  1017 hPa
10   10:00  22°C   9 km/h       57%  1017 hPa
11   09:30  22°C   9 km/h       57%  1017 hPa
12   09:00  21°C   7 km/h       60%  1017 hPa
13   08:30  20°C     Calm       64%  1017 hPa
14   08:00  19°C     Calm       68%  1016 hPa
15   07:30  19°C   7 km/h       68%  1016 hPa
16   06:00  19°C   7 km/h       68%  1016 hPa
17   05:30  19°C   7 km/h       73%  1015 hPa
18   05:00  19°C     Calm       78%  1015 hPa
19   04:30  19°C   6 km/h       78%  1015 hPa
20   04:00  19°C   6 km/h       78%  1015 hPa
21   03:30  19°C   7 km/h       78%  1014 hPa
22   03:00  19°C   6 km/h       78%  1014 hPa
23   02:30  19°C     Calm       78%  1014 hPa
```

10)MONUMENT

```
In [245]: from bs4 import BeautifulSoup
          import requests
          import pandas as pd
```

```
In [246]: page=requests.get("https://www.puredestinations.co.uk/top-10-famous-monuments-to-
```

```
In [247]: page
```

```
Out[247]: <Response [200]>
```

In [248]:
```
soup=BeautifulSoup(page.text,'html.parser')
```

In [249]:
```
print(soup.prettify())
```

...

In [250]:
```
data=[]
```

In [251]:
```
data=soup.find_all("div",class_="blog--single__content column--3-4 u-spacing-thir
```

In [252]:

```python
for i in data:
    print(''.join(i.find_all(text=True)))
```

```
Top 10 Famous Monuments to Visit In India
Rich in culture and diversity, India is home to some of the finest historical m
onuments in the world. Most recognised by the UNESCO World Heritage Site, the f
amous Indian monuments include the beautiful Taj Mahal, the sacred Golden Templ
e and the cultural site, Hawa Mahal. Discover and experience the magnificent in
sights into India’s rich heritage and ancient architecture. Read on for our lis
t of the top must see historical monuments in India below.
Taj Mahal, Agra
Enlisted in the Seven Wonders of the World, The Taj Mahal is one of the most be
autiful and famous buildings located in the city of Agra. This white marble mon
ument was built by a Mughal Emperor called Shahajahan in memory of his beloved
wife. Due to its amazing architecture and the history behind it, this world her
itage site has become very popular to visit by all travellers and romantics fro
m all over the world.

Golden Temple (Harmandir Sahib), Amritsar
The holiest shrine and pilgrimage place located in Amritsar is The Golden Templ
e known as the Harmandir Sahib. This is the most famous and sacred Sikh Gurdwar
a in Punjab, India, adorned with rich history and gold gilded exterior. If you
are interested in culture and history, be sure to visit this popular attraction
in India.

Meenakshi Temple, Madurai
Meenakshi Temple is situated on the Southern banks of Vaigai River in the templ
e city Madurai. This temple is dedicated to Parvati and her consort, Shiva and
is visited by most Hindu and Tamil devotees and architectural lovers throughout
the world. It is believed that this shrine houses 33,000 sculptures in its 14 g
opurams. It’s no doubt one place to visit if you are impressed with art and cul
tural history.

Mysore Palace, Mysore
The Mysore Palace is a famous historical monument in the city of Mysore in Karn
ataka. Commonly described as the City of Palaces, this is the most famous touri
st attraction in India after the Taj Mahal. It is a sight not be missed with it
s spacious halls, lovely art paintings and Indo-Saracenic style architecture. B
est time to visit is at night due to the astonishing illuminated lights coverin
g the whole monument.

Gateway of India, Mumbai
Even though Mumbai is famous for its Bollywood actors and movies, the most famo
us attraction in Mumbai is The Gateway of India. It is a popular gathering spot
for locals, travellers, street vendors and photographers and is known as the Ta
j Mahal of Mumbai. The majestic monument was built to commemorate the visit of
King George V and Queen Mary to Bombay. With so much fun and excitement this pl
ace is not to be missed with family or on your tour of India.

Red Fort, New Delhi
Declared as the UNESCO’s World Heritage Site, Red Fort is located in the centre
of beautiful Delhi. If you love learning about history and culture then this fa
mous historic monument is a must see place to visit. Built by the Mughal Empero
r, Shah Jahan in 1648, and housing a number of museums, its walls are built of
red sandstone. The best time to visit is on Independence Day where the Prime Mi
nister of India hoists the national flag at the Red Fort. End the day by headin
g to an Indian restaurant and enjoy the varieties of wonderful cuisines.

Hawa Mahal, Jaipur
Explore a blend of beauty and Rajasthan culture, the Hawa Mahal also known as P
alace of Winds is situated in the capital of Rajasthan, Jaipur. Built from red
```

```
and pink sandstones by the Maharaja Sawi Pratap Singh in 1799, this unique five
storey structure is one of the most prominent tourist attractions in the Jaipur
city.

Qutub Minar, New Delhi
Discover one of the tallest towers in the world and the second tallest Minar of
India standing elegantly in the Capital city, New Delhi. Standing at 72.5 metre
s and consisting of around 379 stairs, this famous monument represents the rich
architecture of India. As it is a UNESCO World Heritage Site made of red sandst
one and decorated with Arabic and Brahmi inscriptions, travellers from around t
he world come to view this most famous tower in India.

Sanchi Stupa, Sanchi
The beautiful and massive dome, Sanchi Stupa also known as the Great Stupa is a
world renowned Buddhist monument in Sanchi, India. It was constructed by Empero
r Ashoka, and is one of the oldest stone structures in the heart of India. Expe
rience the Indian culture by visiting these major attractions in Sanchi includi
ng a number of Buddhist Stupas, monasteries and temples.

Charminar, Hyderabad
No visit to Hyderabad should be complete without visiting the most famous and m
ajestic monument known as the Charminar. This magnificent and striking mosque c
onstructed in 1591 has four minarets and is the most recognisable symbol in the
city of Hyderabad.

Things to know before planning your trip to India

When travelling to India it’s important to make sure you are up-to-date with tr
avel requirements including obtaining an India visa.
It takes around 9 hours and 42 minutes to fly to India from the UK.
Bring your credit or debit card, but rely on cash to pay for most things. Indi
a’s currency is the Indian Rupee.
The best time to visit India is between October and March.

Are you inspired to plan an incredible trip to India? Get in touch with Pure De
stinations today to speak to a travel expert to plan a tailor made itinerary. F
ancy exploring these famous monuments in India? Why not contact us today!
```

```
In [253]: data
```

```
Out[253]: [<div class="blog--single__content column--3-4 u-spacing-third">
 <h3 class="title title--heading">Top 10 Famous Monuments to Visit In India</h3
>
 <p>Rich in culture and diversity, India is home to some of the finest historic
al monuments in the world. Most recognised by the UNESCO World Heritage Site, t
he famous Indian monuments include the beautiful Taj Mahal, the sacred Golden T
emple and the cultural site, Hawa Mahal. Discover and experience the magnificen
t insights into <a href="https://www.puredestinations.co.uk/destinations/asia/i
ndia/" rel="noopener" target="_blank">India</a>’s rich heritage and ancient arc
hitecture. Read on for our list of the top must see historical monuments in Ind
ia below.</p>
 <p><strong>Taj Mahal, Agra</strong></p>
 <p>Enlisted in the Seven Wonders of the World, The Taj Mahal is one of the mos
t beautiful and famous buildings located in the city of Agra. This white marble
monument was built by a Mughal Emperor called Shahajahan in memory of his belov
ed wife. Due to its amazing architecture and the history behind it, this world
heritage site has become very popular to visit by all travellers and romantics
from all over the world.</p>
 <p><img alt="taj-mahal-pd-blog" class="alignnone size-full wp-image-36626 lazy
```

```
In [ ]:
```