

1. Write a python program to display all the header tags from 'en.wikipedia.org/wiki/Main_Page'.

In [1]:

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

In [2]:

```
page = requests.get('https://en.wikipedia.org/wiki/Main_Page')
page
```

Out[2]:

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

In [3]:

```
soup=BeautifulSoup(page.content)
soup
print(soup.prettify())

<!DOCTYPE html>
<html class="client-nojs" dir="ltr" lang="en">
<head>
  <meta charset="utf-8"/>
  <title>
    Wikipedia, the free encyclopedia
  </title>
  <script>
    document.documentElement.className="client-js";RLCONF={"wgBreakFrame
s":!1,"wgSeparatorTransformTable":[],"wgDigitTransformTable":
[],"wgDefaultDateFormat":"dmy","wgMonthNames":[],"January","Februa
ry","March","April","May","June","July","August","September","Octobe
r","November","December"],"wgRequestId":"1b0e59ff-c6a9-4327-b86d-5f0e5c9
a852c","wgCSPNonce":!1,"wgCanonicalNamespace":"","wgCanonicalSpecialPage
Name":!1,"wgNamespaceNumber":0,"wgPageName":"Main_Page","wgTitle":"Main
Page","wgCurRevisionId":1004593520,"wgRevisionId":1004593520,"wgArticleI
d":15580374,"wgIsArticle":!0,"wgIsRedirect":!1,"wgAction":"view","wgUser
Name":null,"wgUserGroups":["*"],"wgCategories":[],"wgPageContentLanguag
e":"en","wgPageContentModel":"wikitext","wgRelevantPageName":"Main_Pag
e"}
```

In [4]:

```
headers=soup.find_all(['h1','h2','h3','h4','h5','h6'])  
headers
```

Out[4]:

```
[<h1 class="firstHeading" id="firstHeading">Main Page</h1>,<h2 class="mp-h2" id="mp-tfa-h2"><span id="From_today.27s_featured_article"></span><span class="mw-headline" id="From_today's_featured_article">From today's featured 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-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_Wikipedia">Other areas of Wikipedia</span></h2>,<h2 class="mp-h2" id="mp-sister"><span id="Wikipedia.27s_sister_projects"></span><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_languages">Wikipedia languages</span></h2>,<h2>Navigation menu</h2>,<h3 class="vector-menu-heading" id="p-personal-label"><span>Personal tools</span></h3>,<h3 class="vector-menu-heading" id="p-namespaces-label"><span>Namespaces</span></h3>,<h3 class="vector-menu-heading" id="p-variants-label"><span>Variants</span></h3>,<h3 class="vector-menu-heading" id="p-views-label"><span>Views</span></h3>,<h3 class="vector-menu-heading" id="p-actions-label"><span>More</span></h3>,<h3><label for="searchInput">Search</label></h3>,<h3 class="vector-menu-heading" id="p-navigation-label"><span>Navigation</span></h3>,<h3 class="vector-menu-heading" id="p-interaction-label"><span>Contribute</span></h3>,<h3 class="vector-menu-heading" id="p-tb-label"><span>Tools</span></h3>,<h3 class="vector-menu-heading" id="p-coll-print_export-label"><span>Print/export</span></h3>,<h3 class="vector-menu-heading" id="p-wikibase-otherprojects-label"><span>In other projects</span></h3>,
```

```
<h3 class="vector-menu-heading" id="p-lang-label">
<span>Languages</span>
</h3>]
```

In []:

In []:

2. Write a python program to display IMDB's Top rated 100 movies' data (i.e. Name, IMDB rating, Year of release) and make data frame.

In [5]:

```
from bs4 import BeautifulSoup
import requests
```

In [6]:

```
page = requests.get('https://www.imdb.com/chart/top/?ref_=nv_mv_250')
page
```

Out[6]:

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

In [7]:

```
soup=BeautifulSoup(page.content)
soup
print(soup.prettify())

<!DOCTYPE html>
<html xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://ogp.me/ns#">
<head>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta content="app-id=342792525, app-argument=imdb:///?src=mdot" name="apple-itunes-app"/>
<style>
body#styleguide-v2 {
    background: no-repeat fixed center top #000;
}
</style>
<script type="text/javascript">
var IMDbTimer={starttime: new Date().getTime(),pt:'java'};
</script>
<script>
if (typeof uet == 'function') {
    uet("bb", "LoadTitle", {wb: 1});
}
```

In [8]:

```
names=soup.find_all('td',class_='titleColumn')
names
```

Out[8]:

```
[<td class="titleColumn">
 1.
 <a href="/title/tt0111161/" title="Frank Darabont (dir.), Tim Rob
 bins, Morgan Freeman">The Shawshank Redemption</a>
 <span class="secondaryInfo">(1994)</span>
</td>,
<td class="titleColumn">
 2.
 <a href="/title/tt0068646/" title="Francis Ford Coppola (dir.), M
 arlon Brando, Al Pacino">The Godfather</a>
 <span class="secondaryInfo">(1972)</span>
</td>,
<td class="titleColumn">
 3.
 <a href="/title/tt0071562/" title="Francis Ford Coppola (dir.), A
 l Pacino, Robert De Niro">The Godfather: Part II</a>
 <span class="secondaryInfo">(1974)</span>
</td>.
```

In [9]:

```
name=[]
for i in range(0,len(names)):
    name.append(names[i].get_text())
name
```

Out[9]:

```
['\n      1.\n      The Shawshank Redemption\n(1994)\n',
 '\n      2.\n      The Godfather\n(1972)\n',
 '\n      3.\n      The Godfather: Part II\n\n(1974)\n',
 '\n      4.\n      The Dark Knight\n\n(2008)\n',
 '\n      5.\n      12 Angry Men\n\n(1957)\n',
 "\n      6.\n      Schindler's List\n\n(1993)\n",
 '\n      7.\n      The Lord of the Rings: The Return of the King\n\n(2003)\n',
 '\n      8.\n      Pulp Fiction\n\n(1994)\n',
 '\n      9.\n      Il buono, il brutto, il cattivo\n\n(1966)\n',
 '\n      10.\n      The Lord of the Rings: The Fellowship of the Ring\n\n(2001)\n',
 '\n      11.\n      Fight Club\n\n(1999)\n',
 '\n      12.\n      Forrest Gump\n\n(1994)\n',
 '\n      13.\n      Inception\n\n(2010)\n',
 '\n      14.\n      The Lord of the Rings: The Two Towers\n\n(2002)\n',
 '\n      15.\n      Star Wars: Episode V - The Empire Strikes Back\n\n(1980)\n']
```

In [10]:

```
name=[s.replace('\n','') for s in name]  
name
```

Out[10]:

```
[ ' 1. The Shawshank Redemption(1994)',  
' 2. The Godfather(1972)',  
' 3. The Godfather: Part II(1974)',  
' 4. The Dark Knight(2008)',  
' 5. 12 Angry Men(1957)',  
" 6. Schindler's List(1993)",  
' 7. The Lord of the Rings: The Return of the King(2003)',  
' 8. Pulp Fiction(1994)',  
' 9. Il buono, il brutto, il cattivo(1966)',  
' 10. The Lord of the Rings: The Fellowship of the Ring(200  
1)',  
' 11. Fight Club(1999)',  
' 12. Forrest Gump(1994)',  
' 13. Inception(2010)',  
' 14. The Lord of the Rings: The Two Towers(2002)',  
' 15. Star Wars: Episode V - The Empire Strikes Back(1980)',  
' 16. The Matrix(1999)',  
' 17. Goodfellas(1990)'.
```

In [11]:

```
ratings=soup.find_all('td',class_='ratingColumn imdbRating')  
ratings
```

Out[11]:

```
[<td class="ratingColumn imdbRating">  
 <strong title="9.2 based on 2,409,927 user ratings">9.2</strong>  
 </td>,  
 <td class="ratingColumn imdbRating">  
 <strong title="9.1 based on 1,667,519 user ratings">9.1</strong>  
 </td>,  
 <td class="ratingColumn imdbRating">  
 <strong title="9.0 based on 1,158,535 user ratings">9.0</strong>  
 </td>,  
 <td class="ratingColumn imdbRating">  
 <strong title="9.0 based on 2,368,730 user ratings">9.0</strong>  
 </td>,  
 <td class="ratingColumn imdbRating">  
 <strong title="8.9 based on 710,200 user ratings">8.9</strong>  
 </td>,  
 <td class="ratingColumn imdbRating">  
 <strong title="8.9 based on 1,241,855 user ratings">8.9</strong>  
 </td>.
```

In [12]:

```
rating=[]
for i in range(0,len(ratings)):
    rating.append(ratings[i].get_text())
rating
```

Out[12]:

```
['\n9.2\n',
 '\n9.1\n',
 '\n9.0\n',
 '\n9.0\n',
 '\n8.9\n',
 '\n8.9\n',
 '\n8.9\n',
 '\n8.8\n',
 '\n8.8\n',
 '\n8.8\n',
 '\n8.8\n',
 '\n8.8\n',
 '\n8.7\n',
 '\n8.7\n',
 '\n8.7\n',
 '\n8.7\n',
 '\n8.6\n',
 '\n8.6\n',
 '\n8.6\n']
```

In [13]:

```
rating=[s.replace('\n','') for s in rating]
rating
```

Out[13]:

```
['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',
 '8.7',
 '8.6',
 '8.6',
 '8.6']
```

In [14]:

```
years=soup.find_all('span',class_='secondaryInfo')
years
```

Out[14]:

```
[<span class="secondaryInfo">(1994)</span>,
 <span class="secondaryInfo">(1972)</span>,
 <span class="secondaryInfo">(1974)</span>,
 <span class="secondaryInfo">(2008)</span>,
 <span class="secondaryInfo">(1957)</span>,
 <span class="secondaryInfo">(1993)</span>,
 <span class="secondaryInfo">(2003)</span>,
 <span class="secondaryInfo">(1994)</span>,
 <span class="secondaryInfo">(1966)</span>,
 <span class="secondaryInfo">(2001)</span>,
 <span class="secondaryInfo">(1999)</span>,
 <span class="secondaryInfo">(1994)</span>,
 <span class="secondaryInfo">(2010)</span>,
 <span class="secondaryInfo">(2002)</span>,
 <span class="secondaryInfo">(1980)</span>,
 <span class="secondaryInfo">(1999)</span>,
 <span class="secondaryInfo">(1990)</span>,
 <span class="secondaryInfo">(1975)</span>.
```

In [15]:

```
year=[]
for i in range(0,len(years)):
    year.append(years[i].get_text())
year
```

Out[15]:

```
['(1994)',
 '(1972)',
 '(1974)',
 '(2008)',
 '(1957)',
 '(1993)',
 '(2003)',
 '(1994)',
 '(1966)',
 '(2001)',
 '(1999)',
 '(1994)',
 '(2010)',
 '(2002)',
 '(1980)',
 '(1999)',
 '(1990)',
 '(1975)']
```

In [16]:

```
import pandas as pd
```

In [17]:

```
df=pd.DataFrame()
df[ 'Titles & Year' ]=name
df[ 'Rating' ]=rating
df[ 'Year' ]=year
```

In [18]:

df

Out[18]:

	Titles & Year	Rating	Year
0	1. The Shawshank Redemption(1994)	9.2	(1994)
1	2. The Godfather(1972)	9.1	(1972)
2	3. The Godfather: Part II(1974)	9.0	(1974)
3	4. The Dark Knight(2008)	9.0	(2008)
4	5. 12 Angry Men(1957)	8.9	(1957)
...
245	246. Trois couleurs: Rouge(1994)	8.0	(1994)
246	247. Drishyam(2013)	8.0	(2013)
247	248. Shin seiki Evangelion Gekijô-b...	8.0	(1997)
248	249. Sunrise: A Song of Two Humans(...	8.0	(1927)
249	250. Miracle in Cell No. 7(2019)	8.0	(2019)

250 rows × 3 columns

In [19]:

df.head(100)

Out[19]:

	Titles & Year	Rating	Year
0	1. The Shawshank Redemption(1994)	9.2	(1994)
1	2. The Godfather(1972)	9.1	(1972)
2	3. The Godfather: Part II(1974)	9.0	(1974)
3	4. The Dark Knight(2008)	9.0	(2008)
4	5. 12 Angry Men(1957)	8.9	(1957)
...
95	96. Jagten(2012)	8.3	(2012)
96	97. Iди и смотри(1985)	8.3	(1985)
97	98. Singin' in the Rain(1952)	8.3	(1952)
98	99. North by Northwest(1959)	8.3	(1959)

In []:

By using another way

In [20]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from unicodedata import normalize

table_Movies = pd.read_html('https://www.imdb.com/chart/top/?ref_=nv_mv_250')
```

In [21]:

```
df = pd.DataFrame()
df
```

Out[21]:

In [22]:

```
print(f'Total tables:{len(table_Movies)})')
```

Total tables:1

In [23]:

```
table_Movies = pd.read_html('https://www.imdb.com/chart/top/?ref_=nv_mv_250')
table_Movies
```

Out[23]:

	Unnamed: 0	Rank & Title \
0	NaN	1. The Shawshank Redemption (1994)
1	NaN	2. The Godfather (1972)
2	NaN	3. The Godfather: Part II (1974)
3	NaN	4. The Dark Knight (2008)
4	NaN	5. 12 Angry Men (1957)
..
245	NaN	246. Trois couleurs: Rouge (1994)
246	NaN	247. Drishyam (2013)
247	NaN	248. Shin seiki Evangelion Gekijô-ban: Air/Ma...
248	NaN	249. Sunrise: A Song of Two Humans (1927)
249	NaN	250. Miracle in Cell No. 7 (2019)

	IMDb Rating	Your Rating	Unnamed: 4
0	9.2	12345678910	NOT YET RELEASED Seen NaN
1	9.1	12345678910	NOT YET RELEASED Seen NaN
2	9.0	12345678910	NOT YET RELEASED Seen NaN
3	9.0	12345678910	NOT YET RFI FASFD Seen NaN

In [24]:

```
df = table_Movies[0]
del df["Unnamed: 0"]
del df["Your Rating"]
del df["Unnamed: 4"]
df.head(100)
```

Out[24]:

	Rank & Title	IMDb Rating
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. Jagten (2012)	8.3
96	97. Iди и смотри (1985)	8.3
97	98. Singin' in the Rain (1952)	8.3
98	99. North by Northwest (1959)	8.3
99	100. Eternal Sunshine of the Spotless Mind (...)	8.3

100 rows × 2 columns

In [25]:

```
df.to_csv('top100_movies.csv')
```

In []:

In []:

3. Write a python program to display IMDb's Top rated 100 Indian movies' data (i.e. Name, IMDb rating, Year of release) and make data frame.

In []:

In []:

In [26]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from unicodedata import normalize

table_imdb_movies = pd.read_html('https://www.imdb.com/india/top-rated-indian-movies/?pf_')
```

In [27]:

```
print(f'Total tables:{len(table_imdb_movies)}')
```

Total tables:1

In [28]:

```
table_imdb_movies = pd.read_html('https://www.imdb.com/india/top-rated-indian-movies/?pf_')
table_imdb_movies
```

Out[28]:

	Unnamed: 0	Rank & Title	IMDb Rating
0	NaN	1. Pather Panchali (1955)	8.5
1	NaN	2. Nayakan (1987)	8.5
2	NaN	3. Pariyerum Perumal (2018)	8.5
3	NaN	4. Anbe Sivam (2003)	8.5
4	NaN	5. Golmaal (1979)	8.5
..
245	NaN	246. Kuch Kuch Hota Hai (1998)	7.5
246	NaN	247. Jodhaa Akbar (2008)	7.5
247	NaN	248. Trapped (2016)	7.5
248	NaN	249. Parmanu: The Story of Pokhran (2018)	7.5
249	NaN	250. Delhi Belly (2011)	7.5
	Your Rating	Unnamed: 4	
0	12345678910	NOT YET RELEASED	Seen
1	12345678910	NOT YET RELEASED	Seen
2	12345678910	NOT YET RELEASED	Seen
3	12345678910	NOT YET RELEASED	Seen
4	12345678910	NOT YET RELEASED	Seen
..	
245	12345678910	NOT YET RELEASED	Seen
246	12345678910	NOT YET RELEASED	Seen
247	12345678910	NOT YET RELEASED	Seen
248	12345678910	NOT YET RELEASED	Seen
249	12345678910	NOT YET RELEASED	Seen

[250 rows x 5 columns]]

In [29]:

```
df = table_imdb_movies[0]
del df["Unnamed: 4"]
del df["Your Rating"]
del df["Unnamed: 0"]
df.head(100)
```

Out[29]:

Rank & Title IMDb Rating

	Rank & Title	IMDb Rating
0	1. Pather Panchali (1955)	8.5
1	2. Nayakan (1987)	8.5
2	3. Pariyerum Perumal (2018)	8.5
3	4. Anbe Sivam (2003)	8.5
4	5. Golmaal (1979)	8.5
...
95	96. Andaz Apna Apna (1994)	8.1
96	97. Virumandi (2004)	8.1
97	98. Uri: The Surgical Strike (2018)	8.1
98	99. PK (2014)	8.1
99	100. Lucia (2013)	8.1

100 rows × 2 columns

In [30]:

df.head(100)

Out[30]:

	Rank & Title	IMDb Rating
0	1. Pather Panchali (1955)	8.5
1	2. Nayakan (1987)	8.5
2	3. Pariyerum Perumal (2018)	8.5
3	4. Anbe Sivam (2003)	8.5
4	5. Golmaal (1979)	8.5
...
95	96. Andaz Apna Apna (1994)	8.1
96	97. Virumandi (2004)	8.1
97	98. Uri: The Surgical Strike (2018)	8.1
98	99. PK (2014)	8.1
99	100. Lucia (2013)	8.1

100 rows × 2 columns

In [31]:

df.head(100).to_csv('top100 indian imovies.csv', index=False, header=True)

In []:

4. Write a python program to scrap book name, author name, genre and book review of any 5 books from 'www.bookpage.com' (<http://www.bookpage.com%E2%80%99>).

In [32]:

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

In [33]:

```
url = 'https://bookpage.com/reviews'
response = requests.get(url)
htmlcontent = response.content
soup = BeautifulSoup(htmlcontent, 'html.parser')
print(soup.prettify)

<bound method Tag.prettify of <!DOCTYPE html>

<html lang="en" xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://opengraphprotocol.org/schema/">
<head>
<title>Book Reviews | BookPage</title>
<meta content="Book recommendations of the best new books by genre and more." name="description"/>
<meta content="book reviews, books and literature, writing and writers" name="keywords"/>
<link href="https://bookpage.com/reviews" rel="canonical"/>
<meta content="BookPage.com" property="og:site_name"/>
<meta content="Book Reviews" property="og:title"/>
<meta content="Book recommendations of the best new books by genre and more." property="og:description"/>
<meta content="website" property="og:type"/>
<meta content="https://bookpage.com/reviews" property="og:url"/>
<meta content="//www.bookpage.com/default_image.jpg" property="og:image"/>
<meta content="https://bookpage.com/reviews" property="og:article:published_time"/>
<meta content="https://bookpage.com/reviews" property="og:article:modified_time"/>
<meta content="https://bookpage.com/reviews" property="og:article:author"/>
<meta content="https://bookpage.com/reviews" property="og:article:publisher"/>
<meta content="https://bookpage.com/reviews" property="og:article:section"/>
<meta content="https://bookpage.com/reviews" property="og:article:tag"/>
```

In [34]:

```
books=soup.find_all('div',class_='bp-block article-info')
books
```

Out[34]:

```
[<div class="bp-block article-info">
 <div class="article-list-left">
 <a href="/reviews/26396-gabi-snyder-listen-childrens"></a>
 </div>
 <div class="flex-article-content">
 <h4 class="italic">
 <a href="/reviews/26396-gabi-snyder-listen-childrens">Listen</a>
 </h4>
 <p class="sans bold">
 Gabi Snyder, Stephanie Graegin
 </p>
 <p class="genre-links hidden-phone">
 <a href="/search?book_genre=children_s">Children's</a>
 /
 <a href="/search?book_genre=children_s_picture_book">Children's Picture Book</a>
 </p>
 <div class="flex-article-content">
 <h4 class="italic">
 <a href="/reviews/26396-gabi-snyder-listen-childrens">Listen</a>
 </h4>
 <p class="sans bold">
 Gabi Snyder, Stephanie Graegin
 </p>
 <p class="genre-links hidden-phone">
 <a href="/search?book_genre=children_s">Children's</a>
 /
 <a href="/search?book_genre=children_s_picture_book">Children's Picture Book</a>
 </p>
 </div>
 </div>
 </div>
```

In [35]:

```
book=[]
for i in range(0,len(books)):
    book.append(books[i].get_text())
book
```

Out[35]:

["\n\n\n\n\nListen\n\n\nGabi Snyder, Stephanie Graegin\n\nChildren's\n / \nChildren's Picture Book\n\nIn the opening pages of Listen, a girl stands on her front porch, her backpack resting squarely on her shoulders. She is surrounded by the din of the city.\n\nRead the Review \n\n", '\n\n\n\n\n★ Rise to the Sun\n\nLeah Johnson\n\nYA\n / \nYA Fiction\n\nLeah Johnson's second novel, Rise to the Sun, is filled with evocative details of the music festival experience and moving descriptions of live music.\n\nRead the Review \n\n', '\n\n\n\n\nFilthy Animals\n\nBrandon Taylor\n\nFiction\n / \nShort Stories\n\nIn Brandon Taylor's short story collection, sexual tension acts like an undertow, lurking to pull its victims down below.\n\nRead the Review \n\n', "\n\n\n\n\nThe Woman They Could Not Silence\n\nKate Moore\n\nNonfiction\n / \nHistory\n / \nWomen's History\n\nIn the 19th century, a brave, brilliant and completely healthy woman named Elizabeth Packard was involuntarily committed to an Illinois asylum by her husband.\n\nRead the Review \n\n", '\n\n\n\n\nUnder a Dancing Star\n\nLaura Wood\n\nYA\n / \nYA Fiction\n\nUnder a Dancing Star is an effervescent retelling of Much Ado About Nothing, in which author Laura Wood transplants Shakespeare's Beatrice and Benedick to an artists' colony in Tuscany.\n\nRead the Review \n\n', '\n\n\n\n\nYoke\n\nJessamyn Stanley\n\nNonfiction\n / \nEssays\n / \nBody, Mind & Spirit\n\nThe 13 autobiographical essays in Yoke are brash, outspoken, funny, insightful, honest and occasionally spiced with dashes of self-deprecating melodrama.\n\nRead the Review \n\n', '\n\n\n\n\nLeaving Isn't the Hardest Thing\n\nLauren Hough, Cate Blanchett\n\nAudio\n / \nNonfiction\n / \nEssays\n / \nAuthor Lauren Hough and actor-producer Cate Blanchett create a heartbreakingly intimate experience for listeners.\n\nRead the Review \n\n', '\n\n\n\n\nThe Rose Code\n\nKate Quinn, Saskia Maarleveld\n\nAudio\n / \nFiction\n / \nHistorical Fiction\n\nThe Rose Code is a terrific story, brilliantly performed by Saskia Maarleveld. Or as Osla would say, it's a real corker!\n\nRead the Review \n\n', '\n\n\n\n\nThe Damage\n\nCaitlin Wahrer\n\nMystery & Suspense\n / \nSuspense\n\nWith twists worthy of a season finale of "Law & Order: SVU," The Damage explores a family's struggle in the aftermath of a violent sexual assault.\n\nRead the Review \n\n', '\n\n\n\n\n★ Blackout\n\nDhonielle Clayton, Tiffany D. Jackson, Nic Stone, Angie Thomas, Ashley Woodfolk, Nicola Yoon\n\nYA\n / \nYA Fiction\n\nIn Blackout, six of YA's biggest superstars join forces to create a memorable collection of interlinked love stories that all unfold on one unforgettable New York City night.\n\nRead the Review \n\n"]

In [36]:

```
book=[s.replace('\n\n\n','') for s in book]
book=[s.replace('\n','') for s in book]
book=[s.replace('★','') for s in book]
book
```

Out[36]:

["ListenGabi Snyder, Stephanie GraeginChildren's / Children's Picture Book
In the opening pages of Listen, a girl stands on her front porch, her back pack resting squarely on her shoulders. She is surrounded by the din of th e city.Read the Review ",
' Rise to the SunLeah JohnsonYA / YA FictionLeah Johnson's second novel, Rise to the Sun, is filled with evocative details of the music festival ex perience and moving descriptions of live music.Read the Review ',
'Filthy AnimalsBrandon TaylorFiction / Short StoriesIn Brandon Taylor's s hort story collection, sexual tension acts like an undertow, lurking to pu ll its victims down below.Read the Review ',
"The Woman They Could Not SilenceKate MooreNonfiction / History / Women's HistoryIn the 19th century, a brave, brilliant and completely healthy woma n named Elizabeth Packard was involuntarily committed to an Illinois asylu m by her husband.Read the Review ",
'Under a Dancing StarLaura WoodYA / YA FictionUnder a Dancing Star is an effervescent retelling of Much Ado About Nothing, in which author Laura Wo od transplants Shakespeare's Beatrice and Benedick to an artists' colony i n Tuscany.Read the Review ',
'YokeJessamyn StanleyNonfiction / Essays / Body, Mind & SpiritThe 13 auto biographical essays in Yoke are brash, outspoken, funny, insightful, honest and occasionally spiced with dashes of self-deprecating melodrama.Read t he Review ',
'Leaving Isn't the Hardest ThingLauren Hough, Cate BlanchettAudio / Nonfi ction / EssaysAuthor Lauren Hough and actor-producer Cate Blanchett create a heartbreakin and intimate experience for listeners.Read the Review ',
'The Rose CodeKate Quinn, Saskia MaarleveldAudio / Fiction / Historical F ictionThe Rose Code is a terrific story, brilliantly performed by Saskia M aarleveld. Or as Osla would say, it's a real corker!Read the Review ',
'The DamageCaitlin WahrerMystery & SuspenseWith twists worthy of a season finale of "Law & Order: SVU," The Damage explores a family's s truggle in the aftermath of a violent sexual assault.Read the Review ',
' BlackoutDhonielle Clayton, Tiffany D. Jackson, Nic Stone, Angie Thoma s, Ashley Woodfolk, Nicola YoonYA / YA FictionIn Blackout, six of YA's big gest superstars join forces to create a memorable collection of interlinke d love stories that all unfold on one unforgettable New York City night.Re ad the Review ']

In [37]:

```
len(book)
```

Out[37]:

10

In [38]:

```
authors=soup.find_all('p',class_='sans bold')
authors
```

Out[38]:

```
[<p class="sans bold">
Gabi Snyder, Stephanie Graegin
</p>,
<p class="sans bold">
Leah Johnson
</p>,
<p class="sans bold">
Brandon Taylor
</p>,
<p class="sans bold">
Kate Moore
</p>,
<p class="sans bold">
Laura Wood
</p>,
<p class="sans bold">
Jessamyn Stanley
</p>,
<p class="sans bold">
Lauren Hough, Cate Blanchett
</p>,
<p class="sans bold">
Kate Quinn, Saskia Maarleveld
</p>,
<p class="sans bold">
Caitlin Wahrer
</p>,
<p class="sans bold">
Dhonielle Clayton, Tiffany D. Jackson, Nic Stone, Angie Thomas, Ashley Wo
odfolk, Nicola Yoon
</p>]
```

In [39]:

```
author=[]
for i in range(0,len(authors)):
    author.append(authors[i].get_text())
author
```

Out[39]:

```
['\nGabi Snyder, Stephanie Graegin\n',
'\nLeah Johnson\n',
'\nBrandon Taylor\n',
'\nKate Moore\n',
'\nLaura Wood\n',
'\nJessamyn Stanley\n',
'\nLauren Hough, Cate Blanchett\n',
'\nKate Quinn, Saskia Maarleveld\n',
'\nCaitlin Wahrer\n',
'\nDhonielle Clayton, Tiffany D. Jackson, Nic Stone, Angie Thomas, Ashley
Woodfolk, Nicola Yoon\n']
```

In [40]:

```
author=[s.replace('\n','') for s in author]  
author
```

Out[40]:

```
['Gabi Snyder, Stephanie Graegin',  
'Leah Johnson',  
'Brandon Taylor',  
'Kate Moore',  
'Laura Wood',  
'Jessamyn Stanley',  
'Lauren Hough, Cate Blanchett',  
'Kate Quinn, Saskia Maarleveld',  
'Caitlin Wahrer',  
'Dhonielle Clayton, Tiffany D. Jackson, Nic Stone, Angie Thomas, Ashley Woodfolk, Nicola Yoon']
```

In [41]:

```
len(authors)
```

Out[41]:

```
10
```

In [42]:

```
genres=soup.find_all('p',class_='genre-links hidden-phone')
genres
```

Out[42]:

```
<p class="genre-links hidden-phone">
<a href="/search?book_genre=children_s">Children's</a>
/
<a href="/search?book_genre=children_s_picture_book">Children's Picture
Book</a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=ya">YA</a>
/
<a href="/search?book_genre=ya_fiction">YA Fiction</a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=fiction">Fiction</a>
/
<a href="/search?book_genre=short_stories">Short Stories</a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=nonfiction">Nonfiction</a>
/
<a href="/search?book_genre=history">History</a>
/
<a href="/search?book_genre=women_s_history">Women's History</a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=ya">YA</a>
/
<a href="/search?book_genre=ya_fiction">YA Fiction</a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=nonfiction">Nonfiction</a>
/
<a href="/search?book_genre=essays">Essays</a>
/
<a href="/search?book_genre=body_mind_spirit">Body, Mind & Spirit</
a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=audio">Audio</a>
/
<a href="/search?book_genre=nonfiction">Nonfiction</a>
/
<a href="/search?book_genre=essays">Essays</a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=audio">Audio</a>
/
<a href="/search?book_genre=fiction">Fiction</a>
/
<a href="/search?book_genre=historical_fiction">Historical Fiction</a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=mystery_suspense">Mystery & Suspense</a
>
/
```

```
<a href="/search?book_genre=suspense">Suspense</a>
</p>,
<p class="genre-links hidden-phone">
<a href="/search?book_genre=ya">YA</a>
/
<a href="/search?book_genre=ya_fiction">YA Fiction</a>
</p>]
```

In [43]:

```
genre=[]
for i in range(0,len(genres)):
    genre.append(genres[i].get_text())
genre
```

Out[43]:

```
["\nChildren's\n / \nChildren's Picture Book",
'\nYA\n / \nYA Fiction\n',
'\nFiction\n / \nShort Stories\n',
"\nNonfiction\n / \nHistory\n / \nWomen's History\n",
'\nYA\n / \nYA Fiction\n',
'\nNonfiction\n / \nEssays\n / \nBody, Mind & Spirit\n',
'\nAudio\n / \nNonfiction\n / \nEssays\n',
'\nAudio\n / \nFiction\n / \nHistorical Fiction\n',
'\nMystery & Suspense\n / \nSuspense\n',
'\nYA\n / \nYA Fiction\n']
```

In [44]:

```
genre=[s.replace('\n','') for s in genre]
genre
```

Out[44]:

```
["Children's / Children's Picture Book",
'YA / YA Fiction',
'Fiction / Short Stories',
"Nonfiction / History / Women's History",
'YA / YA Fiction',
'Nonfiction / Essays / Body, Mind & Spirit',
'Audio / Nonfiction / Essays',
'Audio / Fiction / Historical Fiction',
'Mystery & Suspense / Suspense',
'YA / YA Fiction']
```

In [45]:

```
len(genre)
```

Out[45]:

10

In [46]:

```
reviews=soup.find_all('p',class_='excerpt')
reviews
```

Out[46]:

```
[<p class="excerpt">
 <p>In the opening pages of <b>Listen</b>, <b> </b>a girl stands on her front porch, her backpack resting squarely on her shoulders. She is surrounded by the din of the city.</p>
</p>,
<p class="excerpt">
 <p>Leah Johnson's second novel, <b>Rise to the Sun</b>, is filled with evocative details of the music festival experience and moving descriptions of live music.</p>
</p>,
<p class="excerpt">
 In Brandon Taylor's short story collection, sexual tension acts like an undertow, lurking to pull its victims down below.
</p>,
<p class="excerpt">
 In the 19th century, a brave, brilliant and completely healthy woman named Elizabeth Packard was involuntarily committed to an Illinois asylum by her husband.
</p>,
<p class="excerpt">
 <p><b>Under a Dancing Star </b>is an effervescent retelling of <i>Much Ado About Nothing</i>, in which author Laura Wood transplants Shakespeare's Beatrice and Benedick to an artists' colony in Tuscany.</p>
</p>,
<p class="excerpt">
 The 13 autobiographical essays in <i>Yoke</i> are brash, outspoken, funny, insightful, honest and occasionally spiced with dashes of self-deprecating mélodrama.
</p>,
<p class="excerpt">
 Author Lauren Hough and actor-producer Cate Blanchett create a heartbreakingly intimate experience for listeners.
</p>,
<p class="excerpt">
 <strong>The Rose Code</strong> is a terrific story, brilliantly performed by Saskia Maarleveld. Or as Osla would say, it's a real corker!
</p>,
<p class="excerpt">
 <p>With twists worthy of a season finale of "Law & Order: SVU," <strong>The Damage</strong> explores a family's struggle in the aftermath of a violent sexual assault.</p>
</p>,
<p class="excerpt">
 <p>In <b>Blackout</b>, six of YA's biggest superstars join forces to create a memorable collection of interlinked love stories that all unfold on one unforgettable New York City night.</p>
</p>]
```

In [47]:

```
review=[]
for i in range(0,len(reviews)):
    review.append(reviews[i].get_text())
review
```

Out[47]:

['\nIn the opening pages of Listen, a girl stands on her front porch, her backpack resting squarely on her shoulders. She is surrounded by the din of the city.\n',
 '\nLeah Johnson's second novel, Rise to the Sun, is filled with evocative details of the music festival experience and moving descriptions of live music.\n',
 '\nIn Brandon Taylor's short story collection, sexual tension acts like a n undertow, lurking to pull its victims down below.\n',
 '\nIn the 19th century, a brave, brilliant and completely healthy woman named Elizabeth Packard was involuntarily committed to an Illinois asylum b y her husband.\n',
 '\nUnder a Dancing Star is an effervescent retelling of Much Ado About No thing, in which author Laura Wood transplants Shakespeare's Beatrice and Benedick to an artists' colony in Tuscany.\n',
 '\nThe 13 autobiographical essays in Yoke are brash, outspoken, funny, in sightful, honest and occasionally spiced with dashes of self-deprecating m elodrama.\n',
 '\nAuthor Lauren Hough and actor-producer Cate Blanchett create a heartbr eaking and intimate experience for listeners.\n',
 '\nThe Rose Code is a terrific story, brilliantly performed by Saskia Maa rleveld. Or as Osla would say, it's a real corker!\n',
 '\nWith twists worthy of a season finale of "Law & Order: SVU," The Damage explores a family's struggle in the aftermath of a violent sexual assault.\n',
 '\nIn Blackout, six of YA's biggest superstars join forces to create a memorable collection of interlinked love stories that all unfold on one unforgettable New York City night.\n']

In [48]:

```
review=[s.replace('\n','') for s in review]
review
```

Out[48]:

['In the opening pages of Listen, a girl stands on her front porch, her backpack resting squarely on her shoulders. She is surrounded by the din of the city.',
 'Leah Johnson's second novel, Rise to the Sun, is filled with evocative details of the music festival experience and moving descriptions of live music.',
 'In Brandon Taylor's short story collection, sexual tension acts like an undertow, lurking to pull its victims down below.',
 'In the 19th century, a brave, brilliant and completely healthy woman named Elizabeth Packard was involuntarily committed to an Illinois asylum by her husband.',
 'Under a Dancing Star is an effervescent retelling of Much Ado About Nothing, in which author Laura Wood transplants Shakespeare's Beatrice and Benedick to an artists' colony in Tuscany.',
 'The 13 autobiographical essays in Yoke are brash, outspoken, funny, insightful, honest and occasionally spiced with dashes of self-deprecating mordrama.',
 'Author Lauren Hough and actor-producer Cate Blanchett create a heartrending and intimate experience for listeners.',
 'The Rose Code is a terrific story, brilliantly performed by Saskia Maasveld. Or as Osela would say, it's a real corker!',
 'With twists worthy of a season finale of "Law & Order: SVU," The Damage explores a family's struggle in the aftermath of a violent sexual assault.',
 'In Blackout, six of YA's biggest superstars join forces to create a memorable collection of interlinked love stories that all unfold on one unforgettable New York City night.]

In [49]:

```
print(len(book),len(author),len(genre),len(review))
```

10 10 10 10

In [50]:

```
import pandas as pd
df = pd.DataFrame()
df
```

Out[50]:

—

In [51]:

```
df['Books']=book
df['Authors']=author
df['Genres']=genre
df['Review']=review
```

In [52]:

df

Out[52]:

	Books	Authors	Genres	Review
0	ListenGabi Snyder, Stephanie GraeginChildren's...	Gabi Snyder, Stephanie Graegin	Children's / Children's Picture Book	In the opening pages of Listen, a girl stands ...
1	Rise to the SunLeah JohnsonYA / YA FictionLe...	Leah Johnson	YA / YA Fiction	Leah Johnson's second novel, Rise to the Sun, ...
2	Filthy AnimalsBrandon TaylorFiction / Short St...	Brandon Taylor	Fiction / Short Stories	In Brandon Taylor's short story collection, se...
3	The Woman They Could Not SilenceKate MooreNonf...	Kate Moore	Nonfiction / History / Women's History	In the 19th century, a brave, brilliant and co...
4	Under a Dancing StarLaura WoodYA / YA FictionU...	Laura Wood	YA / YA Fiction	Under a Dancing Star is an effervescent retell...
5	YokeJessamyn StanleyNonfiction / Essays / Body...	Jessamyn Stanley	Nonfiction / Essays / Body, Mind & Spirit	The 13 autobiographical essays in Yoke are bra...
6	Leaving Isn't the Hardest ThingLauren Hough, C...	Lauren Hough, Cate Blanchett	Audio / Nonfiction / Essays	Author Lauren Hough and actor-producer Cate Bl...
7	The Rose CodeKate Quinn, Saskia MaarleveldAudi...	Kate Quinn, Saskia Maarleveld	Audio / Fiction / Historical Fiction	The Rose Code is a terrific story, brilliantly...
8	The DamageCaitlin WahrerMystery & Suspense / S...	Caitlin Wahrer	Mystery & Suspense / Suspense	With twists worthy of a season finale of "Law ...
9	BlackoutDhonielle Clayton, Tiffany D. Jackson, Nic Sto...	Dhonielle Clayton, Tiffany D. Jackson, Nic Sto...	YA / YA Fiction	In Blackout, six of YA's biggest superstars jo...

In [53]:

df.to_csv('bookpage.csv')

In []:

In []:

In []:

5. Write a python program to scrape cricket rankings from 'www.icc-cricket.com' (<http://www.icc-cricket.com>)

cricket.com%). You have to scrape:

i) Top 10 ODI teams in men's cricket along with the records for matches, points and rating.

In [54]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from unicodedata import normalize

table_team = pd.read_html('https://www.icc-cricket.com/rankings/mens/team-rankings/odi')
```

In [55]:

```
print(f'Total tables:{len(table_team)}')
```

Total tables:1

In [56]:

```
table_team = pd.read_html('https://www.icc-cricket.com/rankings/mens/team-rankings/odi')
len(table_team)
```

Out[56]:

1

In [57]:

```
df = table_team[0]
df.head(10)
```

Out[57]:

Pos	Team	T	Matches	M	Points	P	Rating	R
0	1	New Zealand NZ		17	2054		121	
1	2	Australia AUS		25	2945		118	
2	3	India IND		29	3344		115	
3	4	England ENG		27	3100		115	
4	5	South Africa SA		20	2137		107	
5	6	Pakistan PAK		24	2323		97	
6	7	Bangladesh BAN		27	2438		90	
7	8	West Indies WI		27	2222		82	
8	9	Sri Lanka SL		24	1876		78	
9	10	Afghanistan AFG		17	1054		62	

In []:

In []:

ii) Top 10 ODI Batsmen in men along with the records of their team and rating.

In [58]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from unicodedata import normalize

table_batsmen = pd.read_html('https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batsmen')
```

In [59]:

```
print(f'Total tables:{len(table_batsmen)}')

table_batsmen = pd.read_html('https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batsmen')
len(table_batsmen)
```

Total tables:1

Out[59]:

1

In [60]:

```
df = table_batsmen[0]
del df["Career Best Rating"]
df.head(10)
```

Out[60]:

	Pos	Player	Team	Rating
0	1	Babar Azam	PAK	865
1	2	Virat Kohli	IND	857
2	3	Rohit Sharma	IND	825
3	4	Ross Taylor	NZ	801
4	5	Aaron Finch	AUS	791
5	6	Jonny Bairstow	ENG	785
6	7	Fakhar Zaman	PAK	778
7	NaN	Francois du Plessis	SA	778
8	9	David Warner	AUS	773
9	NaN	Shai Hope	WI	773

In []:

iii) Top 10 ODI bowlers along with the records of their team and rating.

In [61]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from unicodedata import normalize

table_Bowlers = pd.read_html('https://www.icc-cricket.com/rankings/mens/player-rankings/cricket/bowling')
```

In [62]:

```
print(f'Total tables:{len(table_Bowlers)}')
```

Total tables:1

In [63]:

```
table_Bowlers = pd.read_html('https://www.icc-cricket.com/rankings/mens/player-rankings/cricket/bowling')
len(table_Bowlers)
```

Out[63]:

1

In [64]:

```
df = table_Bowlers[0]
del df["Career Best Rating"]
df.head(10)
```

Out[64]:

Pos		Player	Team	Rating
0	1	Trent Boult	NZ	737
1	2	Mehedi Hasan	BAN	713
2	3	Mujeeb Ur Rahman	AFG	708
3	4	Matt Henry	NZ	691
4	5	Jasprit Bumrah	IND	690
5	6	Kagiso Rabada	SA	666
6	7	Chris Woakes	ENG	665
7	8	Josh Hazlewood	AUS	660
8	9	Pat Cummins	AUS	646
9	10	Mustafizur Rahman	BAN	645

In []:

In []:

6. Write a python program to scrape cricket rankings from 'www.icc-cricket.com' (<http://www.icc-cricket.com%E2%80%99>). You have to scrape:

- i) Top 10 ODI teams in women's cricket along with the records for matches, points and rating.

In [65]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from unicodedata import normalize

table_women_team =pd.read_html('https://www.icc-cricket.com/rankings/womens/team-ranking')
```

In [66]:

```
print(f'Total tables:{len(table_women_team)}')
```

Total tables:1

In [67]:

```
table_women_team = pd.read_html('https://www.icc-cricket.com/rankings/womens/team-ranking')
len(table_women_team)
```

Out[67]:

1

In [68]:

```
df = table_women_team[0]
df.head(10)
```

Out[68]:

Pos	Team T	Matches M	Points P	Rating R
0	1 Australia AUS	18	2955	164
1	2 South Africa SA	24	2828	118
2	3 England ENG	17	1993	117
3	4 India IND	20	2226	111
4	5 New Zealand NZ	21	1947	93
5	6 West Indies WI	12	1025	85
6	7 Pakistan PAK	15	1101	73
7	8 Bangladesh BAN	5	306	61
8	9 Sri Lanka SL	11	519	47
9	10 Ireland IRE	2	25	13

In []:

In []:

ii) Top 10 women's ODI players along with the records of their team and rating.

In [69]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from unicodedata import normalize

table_women_batting =pd.read_html('https://www.icc-cricket.com/rankings/womens/player-ran
```

In [70]:

```
print(f'Total tables:{len(table_women_batting)})')
```

Total tables:1

In [71]:

```
table_women_batting = pd.read_html('https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batting')  
table_women_batting
```

Out[71]:

		Pos	Player
r	Team \		
0		1	Tammy Beaumont
t	ENG	2	Lizelle Lee
e	SA	3	Alyssa Healy
y	AUS	4	Stafanie Taylor
r	WI	5	Meg Lanning
g	AUS
..		...	
...	...	95	This player has moved down in the rankings... Sophie Ecclestone
e	ENG	96	This player has moved down in the rankings... Diana Bai
s	PAK	97	

In [72]:

```
df = table_women_batting[0]  
del df['Career Best Rating']  
df.head(10)
```

Out[72]:

	Pos	Player	Team	Rating	
0	1	Tammy Beaumont	ENG	765	
1	2	Lizelle Lee	SA	758	
2	3	Alyssa Healy	AUS	756	
3	4	Stafanie Taylor	WI	746	
4	5	Meg Lanning	AUS	723	
5	6	Amy Satterthwaite	NZ	715	
6	7	Smriti Mandhana	IND	710	
7	8	Mithali Raj	IND	709	
8	9	This player has moved up in the rankings si...	Natalie Sciver	ENG	685
9	10	This player has moved up in the rankings s...	Laura Wolvaardt	SA	683

In []:

In []:

iii) Top 10 women's ODI all-rounder along with the records of their team and rating.

In [73]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from unicodedata import normalize

table_women_all_rounder = pd.read_html('https://www.icc-cricket.com/rankings/womens/player-rankings/odi/all-rounder')
```

In [74]:

```
print(f'Total tables:{len(table_women_all_rounder)}')
```

Total tables:1

In [75]:

```
table_women_all_rounder = pd.read_html('https://www.icc-cricket.com/rankings/womens/player-rankings/odi/all-rounder')
table_women_all_rounder
```

Out[75]:

	Pos	Pla
yer \		
0	1	Marizanne K
app		
1 This player has moved down in the rankings sin...		Ellyse Pe
rry		
2	3	Stafanie Tay
lor		
3	4	Natalie Sci
ver		
4	5	Deepti Sha
rma		
5	6	Jess Jonas
sen		
6	7	Ashleigh Gard
ner		
7 8 This player has moved up in the rankings si...		Dane van Niekerk

In [76]:

```
df = table_women_all_rounder[0]
df.head(10)
```

Out[76]:

	Pos	Player	Team	Rating	Career Best Rating
0	1	Marizanne Kapp	SA	418	418 v India, 17/03/2021
1	This player has moved down in the rankings sin...	Ellyse Perry	AUS	418	548 v West Indies, 11/09/2019
2		Stafanie Taylor	WI	410	559 v New Zealand, 10/10/2013
3	4	Natalie Sciver	ENG	349	349 v New Zealand, 28/02/2021
4	5	Deepti Sharma	IND	343	397 v South Africa, 09/10/2019
5	6	Jess Jonassen	AUS	307	308 v West Indies, 11/09/2019
6	7	Ashleigh Gardner	AUS	252	256 v New Zealand, 04/04/2021
7	8 This player has moved up in the rankings si...	Dane van Niekerk	SA	243	421 v Sri Lanka, 11/02/2019
8	9 This player has moved down in the rankings ...	Sophie Devine	NZ	242	305 v Australia, 05/10/2020
9	10 This player has moved down in the rankings...	Amelia Kerr	NZ	236	247 v Australia, 07/04/2021

In [77]:

```
del df['Career Best Rating']
df
```

Out[77]:

	Pos	Player	Team	Rating
0	1	Marizanne Kapp	SA	418
1	This player has moved down in the rankings sin...	Ellyse Perry	AUS	418
2		Stafanie Taylor	WI	410
3		Natalie Sciver	ENG	349
4		Deepti Sharma	IND	343
5		Jess Jonassen	AUS	307
6		Ashleigh Gardner	AUS	252
7	8 This player has moved up in the rankings si...	Dane van Niekerk	SA	243
8	9 This player has moved down in the rankings ...	Sophie Devine	NZ	242
9	10 This player has moved down in the rankings...	Amelia Kerr	NZ	236
10	NaN	Katherine Brunt	ENG	236
11	12	Shikha Pandey	IND	204
12	13	Jhulan Goswami	IND	200
13	14	Heather Knight	ENG	189
14	15	Rumana Ahmed	BAN	179
15	16	Hayley Matthews	WI	176
16	17	Harmanpreet Kaur	IND	167
17	18	Chamari Athapaththu	SL	164
18	19	Sune Luus	SA	153
19	20	Poonam Yadav	IND	151

In []:

In []:

7. Write a python program to scrape details of all the mobile phones under Rs. 20,000 listed on Amazon.in. The scraped data should include Product Name, Price, Image URL and Average Rating.

In []:

SME told to leave this question. He told to attend this question after learning selenium.

In []:

In []:

In []:

8. Write a python program to extract information about the local weather from the National Weather Service website of USA, <https://www.weather.gov/> (<https://www.weather.gov/>) for the city, San Francisco. You need to extract data about 7 day extended forecast display for the city. The data should include period, short description, temperature and description.

In [78]:

```
from bs4 import BeautifulSoup
import requests

page = requests.get("https://forecast.weather.gov/MapClick.php?lat=37.77712000000025&lon=-122.42000000000002")
```

Out[78]:

<Response [200]>

In [79]:

```
page.content
```

Out[79]:

```
b'<!DOCTYPE html>\n<html class="no-js">\n    <head>\n        <!-- Meta -->\n        <meta name="viewport" content="width=device-width">\n        <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" /><title>National Weather Service</title><meta name="DC.title" content="National Weather Service" /><meta name="DC.description" content="NOAA National Weather Service National Weather Service" /><meta name="DC.creator" content="US Department of Commerce, NOAA, National Weather Service" /><meta name="DC.date.created" scheme="ISO8601" content="" /><meta name="DC.language" scheme="DCTERMS.RFC1766" content="EN-US" /><meta name="DC.keywords" content="weather, National Weather Service" /><meta name="DC.publisher" content="NOAA's National Weather Service" /><meta name="DC.contributor" content="National Weather Service" /><meta name="DC.rights" content="http://www.weather.gov/disclaimer.php" /><meta name="rating" content="General" /><meta name="robots" content="index,follow" />\n\n        <!-- Icons -->\n        <link rel="shortcut icon" href="./images/favicon.ico" type="image/x-icon" />\n\n        <!-- CSS -->\n        <link rel="stylesheet" href="css/bootstrap-3.2.0.min.css"/>\n        <link rel="stylesheet" href="css/bootstrap-theme-3.2.0.min.css"/>\n        <link rel="stylesheet"\n
```

In [80]:

```
soup=BeautifulSoup(page.content, 'html.parser')\nsoup
```

Out[80]:

```
<!DOCTYPE html>\n\n<html class="no-js">\n    <head>\n        <!-- Meta -->\n        <meta content="width=device-width" name="viewport"/>\n        <link href="http://purl.org/dc/elements/1.1/" rel="schema.DC"/><title>National Weather Service</title><meta content="National Weather Service" name="DC.title"><meta content="NOAA National Weather Service National Weather Service" name="DC.description"/><meta content="US Department of Commerce, NOAA, National Weather Service" name="DC.creator"/><meta content="" name="DC.date.created" scheme="ISO8601"/><meta content="EN-US" name="DC.language" scheme="DCTERMS.RFC1766"/><meta content="weather, National Weather Service" name="DC.keywords"/><meta content="NOAA's National Weather Service" name="DC.publisher"/><meta content="National Weather Service" name="DC.contributor"/><meta content="http://www.weather.gov/disclaimer.php" name="DC.rights"/><meta content="General" name="rating"/><meta content="index.follow" name="robots"/>
```

In [81]:

```
print(soup.prettify())  
  
<!DOCTYPE html>  
<html class="no-js">  
  <head>  
    <!-- Meta -->  
    <meta content="width=device-width" name="viewport"/>  
    <link href="http://purl.org/dc/elements/1.1/" rel="schema.DC"/>  
    <title>  
      National Weather Service  
    </title>  
    <meta content="National Weather Service" name="DC.title">  
    <meta content="NOAA National Weather Service National Weather Service" name="DC.description"/>  
    <meta content="US Department of Commerce, NOAA, National Weather Service" name="DC.creator"/>  
    <meta content="" name="DC.date.created" scheme="ISO8601"/>  
    <meta content="EN-US" name="DC.language" scheme="DCTERMS.RFC1766"/>  
    <meta content="weather, National Weather Service" name="DC.keywords"/>  
>  
    <meta content="NOAA's National Weather Service" name="DC.publisher"/>
```

In [82]:

```
seven_day =soup.find(id="seven-day-forecast")  
forecast_item = seven_day.find_all(class_='tombstone-container')  
tonight = forecast_item[0]  
print(tonight.prettify())  
  
<div class="tombstone-container">  
  <p class="period-name">  
    Overnight  
    <br/>  
    <br/>  
  </p>  
  <p>  
      
  </p>  
  <p class="short-desc">  
    Mostly Cloudy  
  </p>  
  <p class="temp temp-low">  
    Low: 55 °F  
  </p>  
</div>
```

In [83]:

```
period = tonight.find(class_='period-name').get_text()
short_desc = tonight.find(class_='short-desc').get_text()
temp = tonight.find(class_='temp').get_text()
print(period)
print(short_desc)
print(temp)
```

Overnight
Mostly Cloudy
Low: 55 °F

In [84]:

```
img = tonight.find('img')
desc = img['title']

print(desc)
```

Overnight: Mostly cloudy, with a low around 55. West southwest wind around 6 mph.

In [85]:

```
period_tags = seven_day.select(".tombstone-container .period-name")
periods = [pt.get_text() for pt in period_tags]
periods
```

Out[85]:

```
['Overnight',
 'Thursday',
 'ThursdayNight',
 'Friday',
 'FridayNight',
 'Saturday',
 'SaturdayNight',
 'Sunday',
 'SundayNight']
```

In [86]:

```

short_descs = [sd.get_text() for sd in seven_day.select(".tombstone-container .short-desc")]
temps = [t.get_text() for t in seven_day.select(".tombstone-container .temp")]
descs = [d["title"] for d in seven_day.select(".tombstone-container img")]
print(short_descs)
print(temps)
print(descs)

['Mostly Cloudy', 'BecomingSunny', 'IncreasingClouds', 'DecreasingClouds',
 'Mostly Cloudy', 'Mostly Sunny', 'Partly Cloudy', 'Mostly Sunny', 'Partly
Cloudy']
['Low: 55 °F', 'High: 68 °F', 'Low: 55 °F', 'High: 67 °F', 'Low: 56 °F',
 'High: 72 °F', 'Low: 56 °F', 'High: 70 °F', 'Low: 57 °F']
['Overnight: Mostly cloudy, with a low around 55. West southwest wind arou
nd 6 mph. ', 'Thursday: Mostly cloudy through mid morning, then gradual cl
earing, with a high near 68. West southwest wind 5 to 10 mph increasing to
11 to 16 mph in the afternoon. Winds could gust as high as 21 mph. ', 'Thu
rsday Night: Increasing clouds, with a low around 55. West southwest wind
7 to 14 mph. ', 'Friday: Mostly cloudy, then gradually becoming sunny, wit
h a high near 67. West southwest wind 7 to 14 mph, with gusts as high as 1
8 mph. ', 'Friday Night: Mostly cloudy, with a low around 56. West southwe
st wind 8 to 16 mph, with gusts as high as 21 mph. ', 'Saturday: Mostly su
nny, with a high near 72.', 'Saturday Night: Partly cloudy, with a low aro
und 56.', 'Sunday: Mostly sunny, with a high near 70.', 'Sunday Night: Par
tly cloudy, with a low around 57.']

```

In [87]:

```

import pandas as pd
weather = pd.DataFrame({
    "Period": periods,
    "Short_desc": short_descs,
    "Temp": temps,
    "Desc": descs
})
weather

```

Out[87]:

	Period	Short_desc	Temp	Desc
0	Overnight	Mostly Cloudy	Low: 55 °F	Overnight: Mostly cloudy, with a low around 55...
1	Thursday	BecomingSunny	High: 68 °F	Thursday: Mostly cloudy through mid morning, t...
2	ThursdayNight	IncreasingClouds	Low: 55 °F	Thursday Night: Increasing clouds, with a low ...
3	Friday	DecreasingClouds	High: 67 °F	Friday: Mostly cloudy, then gradually becoming...
4	FridayNight	Mostly Cloudy	Low: 56 °F	Friday Night: Mostly cloudy, with a low around...
5	Saturday	Mostly Sunny	High: 72 °F	Saturday: Mostly sunny, with a high near 72.
6	SaturdayNight	Partly Cloudy	Low: 56 °F	Saturday Night: Partly cloudy, with a low arou...
7	Sunday	Mostly Sunny	High: 70 °F	Sunday: Mostly sunny, with a high near 70.
8	SundayNight	Partly Cloudy	Low: 57 °F	Sunday Night: Partly cloudy, with a low around...

In []:

In []:

9. Write a python program to scrape fresher job listings from '<https://internshala.com/>' (<https://internshala.com/%E2%80%99>). It should include job title, company name, CTC, and apply date.

In [88]:

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

In [89]:

```
page = requests.get("https://internshala.com/fresher-jobs")
page
```

Out[89]:

<Response [200]>

In [90]:

```
soup=BeautifulSoup(page.content)
soup
print(soup.prettify())
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:fb="https://www.facebook.com/2008/fbml" xmlns:og="http://ogp.me/ns#">
  <head>
    <meta content="IE=9" http-equiv="X-UA-Compatible"/>
    <meta charset="utf-8"/>
    <meta content="width=device-width, initial-scale=1.0 user-scalable=0" name="viewport"/>
    <meta content="272234782795210" property="fb:app_id"/>
    <meta content="article" property="og:type"/>
    <meta content="1200" property="og:image:width"/>
    <meta content="630" property="og:image:height"/>
    <meta content="@Internshala" name="twitter:site"/>
    <meta content="summary_large_image" name="twitter:card"/>
    <meta content="@internshala" name="twitter:creator"/>
    <meta content="#ffffff" name="theme-color"/>
    <meta content="#ffffff" name="msapplication-navbutton-color"/>
    <meta content="telephone=no" name="format-detection"/>
    <link as="font" crossorigin="" href="/static/fonts/Inter.woff2?v=3.11" type="font/woff2"/>
    <link as="font" crossorigin="" href="/static/fonts/Inter.woff?v=3.11" type="font/woff"/>
    <link as="font" crossorigin="" href="/static/fonts/Inter.ttf?v=3.11" type="font/ttf"/>
    <link as="font" crossorigin="" href="/static/fonts/Inter.OTF?v=3.11" type="font/otf"/>
    <link as="font" crossorigin="" href="/static/fonts/Inter.SFNT?v=3.11" type="font/sfnt"/>
    <link as="font" crossorigin="" href="/static/fonts/Inter.EOT?v=3.11" type="font/eot"/>
  </head>
```

In [91]:

```
titles=soup.find_all('div', class_="heading_4_5 profile")
titles
```

Out[91]:

```
[<div class="heading_4_5 profile">
 <a href="/fresher-job/detail/account-executive-field-sales-fresher-jobs
-in-bangalore-hyderabad-at-roado1624515448">Account Executive - Field Sa
les</a> </div>,
<div class="heading_4_5 profile">
 <a href="/fresher-job/detail/remote-software-engineer-fresher-jobs-at-c
rewkarma1624446180">Software Engineer</a> </div>,
<div class="heading_4_5 profile">
 <a href="/fresher-job/detail/remote-associate-software-developer-freshe
r-jobs-at-iqgateway1624444154">Associate Software Developer</a> </div>,
<div class="heading_4_5 profile">
 <a href="/fresher-job/detail/social-media-strategist-and-copywriter-fre
sher-jobs-in-gurgaon-at-internshala1624450750">Social Media Strategist A
nd Copywriter</a> </div>,
<div class="heading_4_5 profile">
 <a href="/fresher-job/detail/level-1-technical-support-engineer-fresher
-jobs-in-pune-at-mithi-software-technologies-private-limited1624444687">
Level 1 Technical Support Engineer</a> </div>.
```

In [92]:

```
job_titles=[]
for i in titles:
    job_titles.append(i.text)
job_titles
```

Out[92]:

```
['\nAccount Executive - Field Sales ',
 '\nSoftware Engineer ',
 '\nAssociate Software Developer ',
 '\nSocial Media Strategist And Copywriter ',
 '\nLevel 1 Technical Support Engineer ',
 '\nAssociate Software Developer ',
 '\nAccounting Assistant ',
 '\nSoftware Engineer ',
 '\nBusiness Development Manager ',
 '\nBusiness Development Executive ',
 '\nProduct Developer - Mathematics ',
 '\nSales Development Representative ',
 '\nFull Stack Engineer ',
 '\nJunior ReactJS Developer ',
 '\nMarketing Manager ',
 '\nProduct Marketing Associate ',
 '\nAdministration Associate ',
 '\nRecruiter ']
```

In [93]:

```
job_titles=[]
for i in titles:
    job_titles.append(i.text.replace("\n", ''))
job_titles
```

Out[93]:

```
['Account Executive - Field Sales ',
 'Software Engineer ',
 'Associate Software Developer ',
 'Social Media Strategist And Copywriter ',
 'Level 1 Technical Support Engineer ',
 'Associate Software Developer ',
 'Accounting Assistant ',
 'Software Engineer ',
 'Business Development Manager ',
 'Business Development Executive ',
 'Product Developer - Mathematics ',
 'Sales Development Representative ',
 'Full Stack Engineer ',
 'Junior ReactJS Developer ',
 'Marketing Manager ',
 'Product Marketing Associate ',
 'Administration Associate ',
 'Recruiter '.
```

In [94]:

```
len(job_titles)
```

Out[94]:

40

In [95]:

```
companies=soup.find_all('a',class_="link_display_like_text")
companies
```

Out[95]:

```
<a class="link_display_like_text" href="/fresher-jobs/fresher-job-at-Ro
aDo">
                    RoaDo
                </a>,
<a class="link_display_like_text" href="/fresher-jobs/fresher-job-at-Cr
ewKarma">
                    CrewKarma
                </a>,
<a class="link_display_like_text" href="/fresher-jobs/fresher-job-at-IQ
Gateway">
                    IQGateway
                </a>,
<a class="link_display_like_text" href="/fresher-jobs/fresher-job-at-In
ternshala">
                    Internshala
                </a>,
<a class="link_display_like_text" href="/fresher-jobs/fresher-job-at-Mi
thi%20Software%20Technologies%20Private%20Limited">
                    Mithi Software Technologies Private Limited
                </a>,
<a class="link_display_like_text" href="/fresher-jobs/fresher-job-at-Wh
ite%20Blink">
```

In [96]:

```
company_names = []
for i in companies:
    company_names.append(i.text.strip())
company_names
```

Out[96]:

```
['RoaDo',
'CrewKarma',
'IQGateway',
'Internshala',
'Mithi Software Technologies Private Limited',
'White Blink',
'Wono Inc',
'Wono Inc',
'Wono Inc',
'Emertxe Information Technologies',
'Open Door Education',
'Content Beta',
'Softway Solutions Private Limited',
'DeepThought Edutech Ventures Private Limited',
'Saltoro Coffee Roasters',
'WebMOBI',
'Global Sun',
'Radish Consultants Private Limited'.
```

In [97]:

```
len(company_names)
```

Out[97]:

40

In [98]:

```
others=soup.find_all('div', class_="internship_other_details_container")
others
```

Out[98]:

```
[<div class="internship_other_details_container">
<div class="other_detail_item_row">
<div class="other_detail_item">
<div class="item_heading">
<i class="ic-16-play-circle"></i>
<span>
                    Start date
                </span>
</div>
<div class="item_body" id="start-date-first">
<span class="start_immediately_mobile">Starts </span>Immediately
                </div>
</div>
<div class="other_detail_item">
<div class="item_heading">
<i class="ic-16-money"></i>
<span>CTC</span>
</div>
```

In [99]:

```
other_details = []
for i in others:
    other_details.append(i.text)
other details
```

Out[99]:

\n\n\nStarts\x00Immediately\n3 - 4 LPA\n\nApply By\nn\n18 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n4.5 - 7 LPA\n\nApply By\nn\n18 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n4.99 - 5 LPA\n\nApply By\nn\n17 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 - 4 LPA\n\nApply By\nn\n17 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 - 3.6 LPA\n\nApply By\nn\n17 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 - 4.5 LPA\n\nApply By\nn\n17 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 - 3.6 LPA\n\nApply By\nn\n17 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 LPA\nly By\nn\n16 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 - 3.25 LPA\n\nApply By\nn\n17 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 LPA\nly By\nn\n16 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 - 3.5 LPA\n\nApply By\nn\n16 Jul' 21\nn\n",\n"\n\n\nStarts\x00Immediately\n3 LPA\nly By\nn\n17 Jul' 21\nn\n"]

In [100]:

```
other_details = []
for i in others:
    other_details.append(i.text.replace("Start date", '').replace("Starts\x00Immediately",
    ""))
other_details
```

Out[100]:

```
["
CTC 3.2 - 4 LPA
y By24 Jul' 21",
"
CTC 3 - 5 LPA
By24 Jul' 21",
"
CTC 3.6 - 10 LPA
ly By24 Jul' 21",
"
CTC 5 - 6.8 LPA
y By24 Jul' 21",
"
CTC 3 LPA
Jul' 21",
"
CTC 3 - 3.6 LPA
v Bv23 Jul' 21".
```

In [101]:

```
[x.strip() for x in other_details if x.strip()]
```

Out[101]:

```
["CTC 3.2 - 4 LPA
ply By24 Jul' 21",
 "CTC 3 - 5 LPA
y By24 Jul' 21",
 "CTC 3.6 - 10 LPA
pply By24 Jul' 21",
 "CTC 5 - 6.8 LPA
ply By24 Jul' 21",
 "CTC 3 LPA
23 Jul' 21",
 "CTC 3 - 3.6 LPA
ply By23 Jul' 21",
 "CTC 5.4 - 6.1 LPA
Apply By23 Jul' 21",
 "CTC 5.5 - 6.3 LPA
Apply By23 Jul' 21",
 "CTC 5.2 - 5.8 LPA
Annlv Bv23 Jul' 21".
```

In [102]:

```
import pandas as pd
jobs=pd.DataFrame({})
jobs['Title']=job_titles
jobs['Company']=company_names
jobs['CTC & Apply By']=other_details
```

In [103]:

jobs

Out[103]:

	Title	Company	CTC & Apply By
0	Account Executive - Field Sales	RoaDo	...
1	Software Engineer	CrewKarma	...
2	Associate Software Developer	IQGateway	...
3	Social Media Strategist And Copywriter	Internshala	...
4	Level 1 Technical Support Engineer	Mithi Software Technologies Private Limited	...
5	Associate Software Developer	White Blink	...
6	Accounting Assistant	Wono Inc	...
7	Software Engineer	Wono Inc	...
8	Business Development Manager	Wono Inc	...

In [104]:

df=pd.DataFrame(jobs)

In [105]:

df

Out[105]:

	Title	Company	CTC & Apply By
0	Account Executive - Field Sales	RoaDo	...
1	Software Engineer	CrewKarma	...
2	Associate Software Developer	IQGateway	...
3	Social Media Strategist And Copywriter	Internshala	...
4	Level 1 Technical Support Engineer	Mithi Software Technologies Private Limited	...
5	Associate Software Developer	White Blink	...
6	Accounting Assistant	Wono Inc	...
7	Software Engineer	Wono Inc	...
8	Business Development Manager	Wono Inc	...

In [106]:

```
df['CTC & Apply By'] = df['CTC & Apply By'].str.replace(" ", "")  
print(df)
```

	Title
0	Account Executive - Field Sales
1	Software Engineer
2	Associate Software Developer
3	Social Media Strategist And Copywriter
4	Level 1 Technical Support Engineer
5	Associate Software Developer
6	Accounting Assistant
7	Software Engineer
8	Business Development Manager
9	Business Development Executive
10	Product Developer - Mathematics
11	Sales Development Representative
12	Full Stack Engineer
13	Junior ReactJS Developer
14	Marketing Manager
15	Product Marketing Associate
16	Administration Associate
17	Recruiter
..	..

In [107]:

df

Out[107]:

	Title	Company	CTC & Apply By
0	Account Executive - Field Sales	RoaDo	CTC3.2-4LPAApplyBy24Jul'21
1	Software Engineer	CrewKarma	CTC3-5LPAApplyBy24Jul'21
2	Associate Software Developer	IQGateway	CTC3.6-10LPAApplyBy24Jul'21
3	Social Media Strategist And Copywriter	Internshala	CTC5-6.8LPAApplyBy24Jul'21
4	Level 1 Technical Support Engineer	Mithi Software Technologies Private Limited	CTC3LPAApplyBy23Jul'21
5	Associate Software Developer	White Blink	CTC3-3.6LPAApplyBy23Jul'21
..

In []:

10. Write a python program to scrape house details from mentioned url. It should include house title, location, area, emi and price

<https://www.nobroker.in/property/sale/bangalore/Electroitype=BHK4&searchParam=W3sibGF0IjoxMi44NDUyMTQ>

(<https://www.nobroker.in/property/sale/bangalore/Electronic%20City?type=BHK4&searchParam=W3sibGF0IjoxMi44NDUyMTQ>)

In [108]:

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

In [109]:

```
url = 'https://www.nobroker.in/property/sale/bangalore/Electronic%20City?type=BHK4&searchParam=W3sibGF0IjoxMi44NDUyMTQ'
response = requests.get(url)
htmlcontent = response.content
soup = BeautifulSoup(htmlcontent, 'html.parser')
print(soup.prettify)
```

```
<bound method Tag.prettify of
<!DOCTYPE html>
<html lang="en"><head>
<meta content="794951570520699" property="fb:pages"/>
<link href="https://www.nobroker.in" rel="canonical"/>
<link href="/assets.nobroker.in/static/img/favicon.png" id="favicon" rel="shortcut icon"/>
<link href="https://images.nobroker.in/static/img/fav64.png" rel="apple-touch-icon"/>
<meta charset="utf-8"/><meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
<meta content="app-id=com.nobroker.app&referrer=utm_source%3Dnobroker%26utm_medium%3DmobileWeb" name="google-play-app"/>
<meta content="app-id=1200507100, app-argument=nobrokerapp://" name="apple-itunes-app"/>
<meta content="#fd3752" name="theme-color"/>
<meta content="4 BHK flats for sale in Electronic City, 4 BHK apartments for sale in Electronic City, houses for sale in Electronic City" name="keywords">
<meta content="4 BHK Flats for Sale in Electronic City, Bangalore" name="description"/>
```

In [110]:

```
flats=soup.find_all('div',class_='nb__2JHK0')
flats
```

Out[110]:

```
[<div class="nb__2JHK0" id="ff8081815ba928cf015ba954b5fb11d0" itemscope="" itemtype="http://schema.org/"> <div class="nb__37AJ0"><div class="nb__sDz3z"><a class="nb__3CnI6" href="/property/buy/4-bhk-apartment-for-sale-in-naganathapura-bangalore/ff8081815ba928cf015ba954b5fb11d0/detail" itemprop="url" target="_blank" title="4 BHK In Independent House For Sale In Naganathapura"><h2 class="heading-6 font-semi-bold nb__1AShY"><span itemprop="name">4 BHK In Independent House For Sale In Naganathapura</span> <svg aria-hidden="true" class="jss1" focusable="false" role="presentation" viewBox="0 0 24 24"><path d="M0 0h24v24H0z" fill="none"></path><path d="M19 19H5V5h7V3H5c-1.11 0-2 .9-2 2v14c0 1.1.89 2 2 2h14c1.1 0 2-.9 2-2v-7h-2v7zM14 3v2h3.591-9.83 9.83 1.41 1.41L19 6.41V10h2V3h-7z"></path></svg></h2></a> <div class="nb__350l7"><div class="nb__2CMjv">Independent House, Doddanagamangala Rd opposite Sobha Silicon Oasis</div><span><a href="/property/buy/4-bhk-apartment-for-sale-in-naganathapura-bangalore/ff8081815ba928cf015ba954b5fb11d0/detail?#explore-nearby" id="exploreNearbuy" target="_blank">Explore Nearby</a></span></div> <span class="displaynone" itemprop="event" itemscope="" item
```

In [111]:

```
flat=[]
for i in range(0,len(flats)):
    flat.append(flats[i].get_text())
flat
```

Out[111]:

['4 BHK In Independent House For Sale In Naganathapura Independent House, Doddanagamangala Rd opposite Sobha Silicon OasisExplore Nearby2019-1 2-244 BHK In Independent House For Sale In NaganathapuraIndependent Ho use, Doddanagamangala Rd opposite Sobha Silicon Oasis4 BHK In Independen t House For Sale In NaganathapuraNaganathapurabangalore5601004 BHK In Independent House For Sale In NaganathapuraNaganathapurabangaloreREADY 560100Title not verifiedNot under loan₹ 1,500 sqftBuiltup₹45,851/MonthEs timated EMI₹80 Lacs₹5,333 per sq.ft.2 PhotosRequest For PhotosNorthFacin g3-5 yearsProperty Age3BathroomsBike and CarParkingGet Owner DetailsNear by: Maruti Concorde Business ParkMetro Cash CarryThalassery Restaurant V elankanni DriveSila RestaurantKonappa Agra'hara',
 "4 BHK For Sale In Daadys Garden In Electronic City Daadys Garden\x00 Kammasandra Rd, Kammasandra, Electronic City, Bengaluru, Karnataka 56010 0, IndiaExplore Nearby2021-04-154 BHK For Sale In Daadys Garden In Elec tronic CityKammasandra Rd, Kammasandra, Electronic City, Bengaluru, Karn ataka 560100, India4 BHK For Sale In Daadys Garden In Electronic CityEl ectronic Citybangalore4 BHK For Sale In Daadys Garden In Electronic Cit yElectronic CitybangaloreREADYTitle not verifiedNot under loan₹ 2,600 sq ftBuiltup₹85,971/MonthEstimated EMI₹1.5 Crores₹5,769 per sq.ft.6 PhotosE astFacing5-10 yearsProperty Age4BathroomsBike and CarParkingGet Owner De tailsNearby: D MartKFCDomino's PizzaTime TravellerHuskur Gate",
 '4 BHK In Independent House For Sale In Sarjapura Independent House, Shantipura Village , S.P Layout , near Shantipura Panchayat OfficeExplo re Nearby2021-06-074 BHK In Independent House For Sale In SarjapuraInd ependent House, Shantipura Village , S.P Layout , near Shantipura Panc hayat Office4 BHK In Independent House For Sale In SarjapuraSarjapurab angalore4 BHK In Independent House For Sale In SarjapuraSarjapurabanga loreREADYTitle not verifiedNot under loan₹ 1,100 sqftBuiltup₹40,120/Mont hEstimated EMI₹70 Lacs₹6,364 per sq.ft.5 PhotosNorthFacing5-10 yearsProp erty Age2BathroomsBike and CarParkingGet Owner DetailsNearby: Maruti Con corde Business ParkD MartSitara Veg RestaurantVolga RestaurantHuskur Gat e',
 '4 BHK Flat For Sale In Electronic City Standalone Building, Shikrip alya, near Shams SchoolExplore Nearby2021-05-234 BHK Flat For Sale In Electronic CityStandalone Building, Shikripalya, near Shams School4 BHK Flat For Sale In Electronic CityElectronic Citybangalore4 BHK Flat Fo r Sale In Electronic CityElectronic CitybangaloreREADYTitle not verifie dUnder Loan₹ 1,400 sqftBuiltup₹20,060/MonthEstimated EMI₹35 Lacs₹2,500 p er sq.ft.7 PhotosEastFacing1-3 yearsProperty Age4BathroomsBike and CarPa rkingGet Owner DetailsNearby: Maruti Concorde Business ParkAdyar Ananda Bhavan Sweets A2B Veg RestaurantParadise BiryaniBarbeque NationAjmera In finity',
 '4 BHK In Independent House For Sale In Electronic City, Independent House, Krishna reddy layout-Near Maruti Suzuki ARENA (Surakshaa Car Care, Bengaluru, Electronic City)Explore Nearby2021-04-044 BHK In Independen t House For Sale In Electronic City, Independent House, Krishna redd y layout-Near Maruti Suzuki ARENA (Surakshaa Car Care, Bengaluru, Electr onic City)4 BHK In Independent House For Sale In Electronic City, Elec tronic City, bangalore4 BHK In Independent House For Sale In Electroni c City, Electronic City, bangaloreREADYTitle not verifiedNot under loan₹ 2,500 sqftBuiltup₹1.43 Lacs/MonthEstimated EMI₹2.5 Crores₹10,000 per sq. ft.1 PhotosRequest For PhotosNorth-EastFacing3-5 yearsProperty Age4Bathr

'oomsBike and CarParkingGet Owner DetailsNearby: Maruti Concorde Business ParkD MartMetro Cash CarryParadise BiryaniKonappana AgraHara',

"4 BHK In Independent House For Sale In Electronic City Independent House, Dhruv Dhamma LayoutNear Heddari IconExplore Nearby2021-04-174 BHK In Independent House For Sale In Electronic CityIndependent House, Dhruv Dhamma LayoutNear Heddari Icon4 BHK In Independent House For Sale In Electronic CityElectronic Citybangalore4 BHK In Independent House For Sale In Electronic CityElectronic CitybangaloreREADYTitle not verifiedNot under loan₹ 2,400 sqftBuiltup₹51,583/MonthEstimated EMI₹90 Lacs₹3,750 per sq.ft.7 PhotosEastFacing3-5 yearsProperty Age3 BathroomsBike and CarParkingGet Owner DetailsNearby: D MartKFCDomino's PizzaTime TravellerHuskur Gate",

'4 BHK Apartment For Sale In Gopalan Gardenia In Electronic City Gopalan Gardenia\xA0 Gopalan gardenia, Veerasandra Main Rd, Veer Sandra, Electronic City, Bengaluru, Karnataka 560100, IndiaExplore Nearby2021-05-18 4 BHK Flat For Sale In Gopalan Gardenia In Electronic CityGopalan gardenia, Veerasandra Main Rd, Veer Sandra, Electronic City, Bengaluru, Karnataka 560100, India4 BHK Flat For Sale In Gopalan Gardenia In Electronic CityElectronic Citybangalore4 BHK Flat For Sale In Gopalan Gardenia In Electronic CityElectronic CitybangaloreREADYTitle not verifiedUnder Loan₹ 2,650 sqftBuiltup₹63,045/MonthEstimated EMI₹1.1 Crores₹4,151 per sq.ft.6 PhotosEastFacing10+ yearsProperty Age3 BathroomsBike and CarParkingGet Owner Details',

'4 BHK Flat For Sale In Electronic City Standalone Building, Konappa AgraHara, Near Saffron SupermarketExplore Nearby2021-05-154 BHK Flat For Sale In Electronic CityStandalone Building, Konappana AgraHara, Near Saffron Supermarket4 BHK Flat For Sale In Electronic CityElectronic Citybangalore4 BHK Flat For Sale In Electronic CityElectronic CitybangaloreREADYTitle not verifiedNot under loan₹ 1,120 sqftBuiltup₹28,657/MonthEstimated EMI₹50 Lacs₹4,464 per sq.ft.2 PhotosRequest For PhotosEastFacing10+ yearsProperty Age2 BathroomsBikeParkingGet Owner DetailsNearby: Maruti Concorde Business ParkD MartMetro Cash CarryParadise BiryaniKonappana AgraHara',

'4 BHK Flat For Sale In Heena Enclave In Electronic City Neeladri Nagar,Near Pioneer Sun BlossomExplore Nearby2021-06-014 BHK Flat For Sale In Heena Enclave In Electronic City Neeladri Nagar,Near Pioneer Sun Blossom4 BHK Flat For Sale In Heena Enclave In Electronic CityElectronic Citybangalore4 BHK Flat For Sale In Heena Enclave In Electronic CityElecronic CitybangaloreREADYTitle not verifiedNot under loan₹ 2,350 sqftBuiltup₹71,643/MonthEstimated EMI₹1.25 Crores₹5,319 per sq.ft.Request For PhotosEastFacingNewly ConstructedProperty Age4 BathroomsBike and CarParkingGet Owner DetailsNearby: Maruti Concorde Business ParkAdyar Ananda Bhavan Sweets A2B Veg RestaurantParadise BiryaniBarbeque NationAjmera Infinity',

"4 BHK In Independent House For Sale In Ananth Nagar Independent House, Glass factory Outlet nd cross Near Grand Mart Explore Nearby2021-04-024 BHK In Independent House For Sale In Ananth NagarIndependent House, Glass factory Outlet nd cross Near Grand Mart 4 BHK In Independent House For Sale In Ananth NagarAnanth Nagarbangalore4 BHK In Independent House For Sale In Ananth NagarAnanth NagarbangaloreREADYTitle not verifiedNot under loan₹ 2,200 sqftBuiltup₹56,168/MonthEstimated EMI₹98 Lacs₹4,455 per sq.ft.11 PhotosEastFacing1-3 yearsProperty Age3 BathroomsBike and CarParkingGet Owner DetailsNearby: Maruti Concorde Business ParkD MartKFCDomino's PizzaHuskur Gate"]

In [112]:

```
flat=[s.replace('4 BHK In Independent House ','')] for s in flat]
flat=[s.replace('4 BHK ','')] for s in flat]
flat
```

Out[112]:

['For Sale In Naganathapura Independent House, Doddanagamangala Rd oppo site Sobha Silicon OasisExplore Nearby2019-12-24For Sale In NaganathapuraIndependent House, Doddanagamangala Rd opposite Sobha Silicon OasisFor Sale In NaganathapuraNaganathapurabangalore560100For Sale In Naganatha puraNaganathapurabangaloreREADY560100Title not verifiedNot under loan₹ 1,500 sqftBuiltup₹45,851/MonthEstimated EMI₹80 Lacs₹5,333 per sq.ft.2 Ph otosRequest For PhotosNorthFacing3-5 yearsProperty Age3BathroomsBike and CarParkingGet Owner DetailsNearby: Maruti Concorde Business ParkMetro Ca sh CarryThalassery Restaurant Velankanni DriveSila RestaurantKonappana A grahara',
 "For Sale In Daadys Garden In Electronic City Daadys Garden\x0 Kammas andra Rd, Kammasandra, Electronic City, Bengaluru, Karnataka 560100, IndiaExplore Nearby2021-04-15For Sale In Daadys Garden In Electronic CityK ammasandra Rd, Kammasandra, Electronic City, Bengaluru, Karnataka 56010 0, IndiaFor Sale In Daadys Garden In Electronic CityElectronic Citybang aloreFor Sale In Daadys Garden In Electronic CityElectronic Citybangalo reREADYTitle not verifiedNot under loan₹ 2,600 sqftBuiltup₹85,971/MonthE stimated FMT₹1.5 Crores₹5.769 per sq.ft.6 PhotosFastFacing5-10 yearsPro

In [113]:

```
len(flats)
```

Out[113]:

10

In [114]:

```
locations=soup.find_all('div',class_='nb__2CMjv')
locations
```

Out[114]:

```
[<div class="nb__2CMjv">Independent House, Doddanagamangala Rd opposite Sobha Silicon Oasis</div>,
 <div class="nb__2CMjv"><a class="nb__17tcx" href="/daadys-garden-electronic-city_bangalore-prjt-5ba009ad714b568105761868" target="_blank">Daadys Garden</a> <!-- -->Kammasandra Rd, Kammasandra, Electronic City, Bengaluru, Karnataka 560100, India</div>,
 <div class="nb__2CMjv">Independent House, Shantipura Village , S.P Layout , near Shantipura Panchayat Office</div>,
 <div class="nb__2CMjv">Standalone Building, Shikaripalya, near Shams School</div>,
 <div class="nb__2CMjv">Independent House, Krishna reddy layout-Near Maruti Suzuki ARENA (Surakshaa Car Care, Bengaluru, Electronic City)</div>,
 <div class="nb__2CMjv">Independent House, Dhruv Dhama LayoutNear Heddari Icon</div>,
 <div class="nb__2CMjv"><a class="nb__17tcx" href="/gopalan-gardenia-electronic-city_bangalore-prjt-5ba009ad714b568105761406" target="_blank">Gopalan Gardenia</a> <!-- -->Gopalan gardenia, Veerasandra Main Rd, Veer Sandra, Electronic City, Bengaluru, Karnataka 560100, India</div>,
 <div class="nb__2CMjv">Standalone Building, Konappana Agrahara, Near Saffron Supermarket</div>,
 <div class="nb__2CMjv"> Neeladri Nagar,Near Pioneer Sun Blossom</div>,
 <div class="nb__2CMjv">Independent House, Glass factory Outlet nd cross N ear Grand Mart </div>]
```

In [115]:

```
location=[]
for i in range(0,len(locations)):
    location.append(locations[i].get_text())
location
```

Out[115]:

```
['Independent House, Doddanagamangala Rd opposite Sobha Silicon Oasis',
 'Daadys Garden\xa0 Kammasandra Rd, Kammasandra, Electronic City, Bengaluru, Karnataka 560100, India',
 'Independent House, Shantipura Village , S.P Layout , near Shantipura Panchayat Office',
 'Standalone Building, Shikaripalya, near Shams School',
 'Independent House, Krishna reddy layout-Near Maruti Suzuki ARENA (Surakshaa Car Care, Bengaluru, Electronic City)',
 'Independent House, Dhruv Dhama LayoutNear Heddari Icon',
 'Gopalan Gardenia\xa0 Gopalan gardenia, Veerasandra Main Rd, Veer Sandra, Electronic City, Bengaluru, Karnataka 560100, India',
 'Standalone Building, Konappana Agrahara, Near Saffron Supermarket',
 ' Neeladri Nagar,Near Pioneer Sun Blossom',
 'Independent House, Glass factory Outlet nd cross Near Grand Mart ']
```

In [116]:

```
len(locations)
```

Out[116]:

10

In [117]:

```
emis=soup.find_all('div',class_='nb__17R6o')
emis
```

Out[117]:

```
[<div class="nb__17R6o"><div class="nb__2NPHR nb__1imT- "><meta content="Title verification and loan availability" itemprop="name"/><span class="nb__2di3N"><span class="nb__3cC_v nb__2fCmT">Title not verified</span><span class="nb__BPQTU nb__1oYW5"><svg aria-hidden="true" class="jss1 nb__3es4W" focusable="false" role="presentation" viewBox="0 0 24 24"><path d="M0 0h24v24H0z" fill="none"></path><g><path d="M9 16.2L4.8 12l-1.4 1.4L9 21 7l-1.4-1.4L9 16.2z"></path></g></svg></span></span><span class="nb__2di3N"><span class="nb__3cC_v nb__2fCmT">Not under loan</span><span class="nb__2w-FV nb__1nquT"><span>₹</span></span></span></div><div class="nb__2NPHR" itemprop="additionalProperty" itemscope="" itemtype="http://schema.org/PropertyValue"><div class="font-semi-bold heading-6" id="minRent" itemprop="valueReference" itemscope="" itemtype="http://schema.org/PropertyValue"><meta content="Minimum Rent" itemprop="name"/><meta content="INR" itemprop="unitCode"/><div class="nb__3oNyC">1,500 sqft</div></div><div class="heading-7">Builtup</div></div><div class="nb__2NPHR"><meta content="RoomTypes Available" itemprop="name"/><div class="font-semi-bold heading-6" id="roomType"><span>₹</span>45,851/Month</div><div class="heading-7">Estimated FMT</div></div><div class="nb__2NPHR" id="mi
```

In [118]:

```
emi=[]
for i in range(0,len(emis)):
    emi.append(emis[i].get_text())
emi
```

Out[118]:

```
['Title not verifiedNot under loan ₹ 1,500 sqft Builtup ₹ 45,851/Month Estimate  
d EMI ₹ 80 Lacs ₹ 5,333 per sq.ft.',  
'Title not verifiedNot under loan ₹ 2,600 sqft Builtup ₹ 85,971/Month Estimate  
d EMI ₹ 1.5 Crores ₹ 5,769 per sq.ft.',  
'Title not verifiedNot under loan ₹ 1,100 sqft Builtup ₹ 40,120/Month Estimate  
d EMI ₹ 70 Lacs ₹ 6,364 per sq.ft.',  
'Title not verifiedUnder Loan ₹ 1,400 sqft Builtup ₹ 20,060/Month Estimated EM  
I ₹ 35 Lacs ₹ 2,500 per sq.ft.',  
'Title not verifiedNot under loan ₹ 2,500 sqft Builtup ₹ 1.43 Lacs/Month Estimate  
d EMI ₹ 2.5 Crores ₹ 10,000 per sq.ft.',  
'Title not verifiedNot under loan ₹ 2,400 sqft Builtup ₹ 51,583/Month Estimate  
d EMI ₹ 90 Lacs ₹ 3,750 per sq.ft.',  
'Title not verifiedUnder Loan ₹ 2,650 sqft Builtup ₹ 63,045/Month Estimated EM  
I ₹ 1.1 Crores ₹ 4,151 per sq.ft.',  
'Title not verifiedNot under loan ₹ 1,120 sqft Builtup ₹ 28,657/Month Estimate  
d EMI ₹ 50 Lacs ₹ 4,464 per sq.ft.',  
'Title not verifiedNot under loan ₹ 2,350 sqft Builtup ₹ 71,643/Month Estimate  
d EMI ₹ 1.25 Crores ₹ 5,319 per sq.ft.',  
'Title not verifiedNot under loan ₹ 2,200 sqft Builtup ₹ 56,168/Month Estimate  
d EMI ₹ 98 Lacs ₹ 4,455 per sq.ft.]
```

In [119]:

```
emi=[s.replace('Title not verifiedNot under loan₹','') for s in emi]  
emi
```

Out[119]:

```
[' 1,500 sqftBuiltup₹45,851/MonthEstimated EMI₹80 Lacs₹5,333 per sq.ft.',  
 ' 2,600 sqftBuiltup₹85,971/MonthEstimated EMI₹1.5 Crores₹5,769 per sq.ft.',  
 ' 1,100 sqftBuiltup₹40,120/MonthEstimated EMI₹70 Lacs₹6,364 per sq.ft.',  
 'Title not verifiedUnder Loan₹ 1,400 sqftBuiltup₹20,060/MonthEstimated EM  
I₹35 Lacs₹2,500 per sq.ft.',  
 ' 2,500 sqftBuiltup₹1.43 Lacs/MonthEstimated EMI₹2.5 Crores₹10,000 per s  
q.ft.',  
 ' 2,400 sqftBuiltup₹51,583/MonthEstimated EMI₹90 Lacs₹3,750 per sq.ft.',  
 'Title not verifiedUnder Loan₹ 2,650 sqftBuiltup₹63,045/MonthEstimated EM  
I₹1.1 Crores₹4,151 per sq.ft.',  
 ' 1,120 sqftBuiltup₹28,657/MonthEstimated EMI₹50 Lacs₹4,464 per sq.ft.',  
 ' 2,350 sqftBuiltup₹71,643/MonthEstimated EMI₹1.25 Crores₹5,319 per sq.f  
t.',  
 ' 2,200 sqftBuiltup₹56,168/MonthEstimated EMI₹98 Lacs₹4,455 per sq.ft.]
```

In [120]:

```
emi=[s.replace('EMI₹80 Lacs₹6,667 per sq.ft.', '') for s in emi]
emi=[s.replace(' 1,200 sqftBuiltup', '') for s in emi]
emi=[s.replace('2,600 sqftBuiltup', '') for s in emi]
emi=[s.replace('1,100 sqftBuiltup', '') for s in emi]
emi=[s.replace('Title not verifiedUnder Loan₹ 1,400 sqftBuiltup', '') for s in emi]
emi=[s.replace('2,500 sqftBuiltup', '') for s in emi]
emi=[s.replace('EMI₹1.5 Crores₹5,769 per sq.ft.', '') for s in emi]
emi=[s.replace('EMI₹70 Lacs₹6,364 per sq.ft.', '') for s in emi]
emi=[s.replace('EMI₹35 Lacs₹2,500 per sq.ft.', '') for s in emi]
emi=[s.replace('EMI₹2.5 Crores₹10,000 per sq.ft.', '') for s in emi]
emi=[s.replace('2,400 sqftBuiltup', '') for s in emi]
emi=[s.replace('EMI₹90 Lacs₹3,750 per sq.ft.', '') for s in emi]
emi=[s.replace('Title not verifiedUnder Loan₹ 2,650 sqftBuiltup', '') for s in emi]
emi=[s.replace('EMI₹1.1 Crores₹4,151 per sq.ft.', '') for s in emi]
emi=[s.replace(' 1,120 sqftBuiltup', '') for s in emi]
emi=[s.replace('EMI₹50 Lacs₹4,464 per sq.ft.', '') for s in emi]
emi=[s.replace('2,350 sqftBuiltup', '') for s in emi]
emi=[s.replace('EMI₹1.25 Crores₹5,319 per sq.ft.', '') for s in emi]
emi=[s.replace('2,200 sqftBuiltup', '') for s in emi]
emi=[s.replace('EMI₹98 Lacs₹4,455 per sq.ft.', '') for s in emi]
emi
```

Out[120]:

```
[ ' 1,500 sqftBuiltup₹45,851/MonthEstimated EMI₹80 Lacs₹5,333 per sq.ft.,
 ' ₹85,971/MonthEstimated ',
 ' ₹40,120/MonthEstimated ',
 ' ₹20,060/MonthEstimated ',
 ' ₹1.43 Lacs/MonthEstimated ',
 ' ₹51,583/MonthEstimated ',
 ' ₹63,045/MonthEstimated ',
 ' ₹28,657/MonthEstimated ',
 ' ₹71,643/MonthEstimated ',
 ' ₹56,168/MonthEstimated ']
```

In [121]:

```
emi=[s.replace('/MonthEstimated', '') for s in emi]
emi
```

Out[121]:

```
[ ' 1,500 sqftBuiltup₹45,851 EMI₹80 Lacs₹5,333 per sq.ft.,
 ' ₹85,971 ',
 ' ₹40,120 ',
 ' ₹20,060 ',
 ' ₹1.43 Lacs ',
 ' ₹51,583 ',
 ' ₹63,045 ',
 ' ₹28,657 ',
 ' ₹71,643 ',
 ' ₹56,168 ']
```

In [122]:

```
areas=soup.find_all('div',class_='nb__17R6o')
areas
```

Out[122]:

```
[<div class="nb__17R6o"><div class="nb__2NPHR nb__1imT- "><meta content="Title verification and loan availability" itemprop="name"/><span class="nb__2di3N"><span class="nb__3cC_v nb__2fCmT">Title not verified</span><span class="nb__BPQTU nb__1oYW5"><svg aria-hidden="true" class="jss1 nb__3es4W" focusable="false" role="presentation" viewBox="0 0 24 24"><path d="M0 0h24v24H0z" fill="none"/></path><g><path d="M9 16.2L4.8 12l-1.4 1.4L9 16.2z"></path></g></svg></span></span><span class="nb__2di3N"><span class="nb__3cC_v nb__2fCmT">Not under loan</span><span class="nb__2w-FV nb__1nquT"><span>₹</span></span></span></div><div class="nb__2NPHR" itemprop="additionalProperty" itemscope="" itemtype="http://schema.org/PropertyValue"><div class="font-semi-bold heading-6" id="minRent" itemprop="valueReference" itemscope="" itemtype="http://schema.org/PropertyValue"><meta content="Minimum Rent" itemprop="name"/><meta content="INR" itemprop="unitCode"/><div class="nb__3oNyC">1,500 sqft</div></div><div class="heading-7">Builtup</div></div><div class="nb__2NPHR"><meta content="RoomTypes Available" itemprop="name"/><div class="font-semi-bold heading-6" id="roomType"><span>₹</span>45,851/Month</div><div class="heading-7">Estimated FMT</div></div><div class="nb__2NPHR" id="mi
```

In [123]:

```
area=[]
for i in range(0,len(areas)):
    area.append(areas[i].get_text())
area
```

Out[123]:

```
['Title not verifiedNot under loan ₹ 1,500 sqft Builtup ₹ 45,851/Month Estimate EMIs ₹ 80 Lacs ₹ 5,333 per sq.ft.', 'Title not verifiedNot under loan ₹ 2,600 sqft Builtup ₹ 85,971/Month Estimate EMIs ₹ 1.5 Crores ₹ 5,769 per sq.ft.', 'Title not verifiedNot under loan ₹ 1,100 sqft Builtup ₹ 40,120/Month Estimate EMIs ₹ 70 Lacs ₹ 6,364 per sq.ft.', 'Title not verifiedUnder Loan ₹ 1,400 sqft Builtup ₹ 20,060/Month Estimated EMIs ₹ 35 Lacs ₹ 2,500 per sq.ft.', 'Title not verifiedNot under loan ₹ 2,500 sqft Builtup ₹ 1.43 Lacs/Month Estimated EMIs ₹ 2.5 Crores ₹ 10,000 per sq.ft.', 'Title not verifiedNot under loan ₹ 2,400 sqft Builtup ₹ 51,583/Month Estimate EMIs ₹ 90 Lacs ₹ 3,750 per sq.ft.', 'Title not verifiedUnder Loan ₹ 2,650 sqft Builtup ₹ 63,045/Month Estimated EMIs ₹ 1.1 Crores ₹ 4,151 per sq.ft.', 'Title not verifiedNot under loan ₹ 1,120 sqft Builtup ₹ 28,657/Month Estimate EMIs ₹ 50 Lacs ₹ 4,464 per sq.ft.', 'Title not verifiedNot under loan ₹ 2,350 sqft Builtup ₹ 71,643/Month Estimate EMIs ₹ 1.25 Crores ₹ 5,319 per sq.ft.', 'Title not verifiedNot under loan ₹ 2,200 sqft Builtup ₹ 56,168/Month Estimate EMIs ₹ 98 Lacs ₹ 4,455 per sq.ft.]
```

In [124]:

```

area=[s.replace('Title not verifiedNot under loan₹', '') for s in area]
area=[s.replace('₹45,851/MonthEstimated EMI₹80 Lacs₹6,667 per sq.ft.', '') for s in area]
area=[s.replace('₹85,971/MonthEstimated EMI₹1.5 Crores₹5,769 per sq.ft.', '') for s in area]
area=[s.replace('₹40,120/MonthEstimated EMI₹70 Lacs₹6,364 per sq.ft.', '') for s in area]
area=[s.replace('₹20,060/MonthEstimated EMI₹35 Lacs₹2,500 per sq.ft.', '') for s in area]
area=[s.replace('₹1.43 Lacs/MonthEstimated EMI₹2.5 Crores₹10,000 per sq.ft.', '') for s in area]
area=[s.replace('₹51,583/MonthEstimated EMI₹90 Lacs₹3,750 per sq.ft.', '') for s in area]
area=[s.replace('₹63,045/MonthEstimated EMI₹1.1 Crores₹4,151 per sq.ft.', '') for s in area]
area=[s.replace('₹28,657/MonthEstimated EMI₹50 Lacs₹4,464 per sq.ft.', '') for s in area]
area=[s.replace('₹71,643/MonthEstimated EMI₹1.25 Crores₹5,319 per sq.ft.', '') for s in area]
area=[s.replace('₹56,168/MonthEstimated EMI₹98 Lacs₹4,455 per sq.ft.', '') for s in area]
area

```

Out[124]:

```

[' 1,500 sqftBuiltup₹45,851/MonthEstimated EMI₹80 Lacs₹5,333 per sq.ft.',
 ' 2,600 sqftBuiltup',
 ' 1,100 sqftBuiltup',
 'Title not verifiedUnder Loan₹ 1,400 sqftBuiltup',
 ' 2,500 sqftBuiltup',
 ' 2,400 sqftBuiltup',
 'Title not verifiedUnder Loan₹ 2,650 sqftBuiltup',
 ' 1,120 sqftBuiltup',
 ' 2,350 sqftBuiltup',
 ' 2,200 sqftBuiltup']

```

In [125]:

```

area=[s.replace('₹45,851 80 ₹6,667 ', '') for s in area]
area=[s.replace('₹85,971 1.5 Crores₹5,769', '') for s in area]
area=[s.replace('₹40,120 70 ₹6,364 ', '') for s in area]
area=[s.replace('₹20,060 35 ₹2,500 ', '') for s in area]
area=[s.replace('₹1.43 2.5 Crores₹10,000 ', '') for s in area]
area=[s.replace('₹51,583 90 ₹3,750 ', '') for s in area]
area=[s.replace('₹63,045 1.1 Crores₹4,151 ', '') for s in area]
area=[s.replace('₹28,657 50 ₹4,464 ', '') for s in area]
area=[s.replace('₹71,643 1.25 Crores₹5,319 ', '') for s in area]
area=[s.replace('₹56,168 98 ₹4,455 ', '') for s in area]
area

```

Out[125]:

```

[' 1,500 sqftBuiltup₹45,851/MonthEstimated EMI₹80 Lacs₹5,333 per sq.ft.',
 ' 2,600 sqftBuiltup',
 ' 1,100 sqftBuiltup',
 'Title not verifiedUnder Loan₹ 1,400 sqftBuiltup',
 ' 2,500 sqftBuiltup',
 ' 2,400 sqftBuiltup',
 'Title not verifiedUnder Loan₹ 2,650 sqftBuiltup',
 ' 1,120 sqftBuiltup',
 ' 2,350 sqftBuiltup',
 ' 2,200 sqftBuiltup']

```

In [126]:

```
area=[s.replace(' sqftBuiltup','')for s in area]
area=[s.replace('₹','')for s in area]
area=[s.replace('Title not verifiedUnder Loan ','')for s in area]
area
```

Out[126]:

```
[' 1,50045,851/MonthEstimated EMI80 Lacs5,333 per sq.ft.',
 ' 2,600',
 ' 1,100',
 '1,400',
 ' 2,500',
 ' 2,400',
 '2,650',
 ' 1,120',
 ' 2,350',
 ' 2,200']
```

In []:

In [127]:

```
prices=soup.find_all('div',class_='font-semi-bold heading-6')
prices
```

Out[127]:

```
[<div class="font-semi-bold heading-6" id="minRent" itemprop="valueReference" itemscope="" itemtype="http://schema.org/PropertyValue"><meta content="Minimum Rent" itemprop="name"/><meta content="INR" itemprop="unitCode"/><div class="nb__3oNyC">1,500 sqft</div></div>,
 <div class="font-semi-bold heading-6" id="roomType"><span>₹</span>45,851/Month</div>,
 <div class="font-semi-bold heading-6"><meta content="Minimum Deposit" itemprop="name"/><span>₹<!-- -->80 Lacs</span></div>,
 <div class="font-semi-bold heading-6" id="minRent" itemprop="valueReference" itemscope="" itemtype="http://schema.org/PropertyValue"><meta content="Minimum Rent" itemprop="name"/><meta content="INR" itemprop="unitCode"/><div class="nb__3oNyC">2,600 sqft</div></div>,
 <div class="font-semi-bold heading-6" id="roomType"><span>₹</span>85,971/Month</div>,
 <div class="font-semi-bold heading-6"><meta content="Minimum Deposit" itemprop="name"/><span>₹<!-- -->1.5 Crores</span></div>,
 <div class="font-semi-bold heading-6" id="minRent" itemprop="valueReference" itemscope="" itemtype="http://schema.org/PropertyValue"><meta content="INR" itemprop="unitCode"/><div class="nb__3oNyC">2,650 sqft</div></div>,
 <div class="font-semi-bold heading-6" id="roomType"><span>₹</span>85,971/Month</div>]
```

In [128]:

```
price=[]
for i in range(0,len(prices)):
    price.append(prices[i].get_text())
price
```

Out[128]:

```
['1,500 sqft',
 '₹45,851/Month',
 '₹80 Lacs',
 '2,600 sqft',
 '₹85,971/Month',
 '₹1.5 Crores',
 '1,100 sqft',
 '₹40,120/Month',
 '₹70 Lacs',
 '1,400 sqft',
 '₹20,060/Month',
 '₹35 Lacs',
 '2,500 sqft',
 '₹1.43 Lacs/Month',
 '₹2.5 Crores',
 '2,400 sqft',
 '₹51,583/Month',
 '₹90 Lacs',
 '2,650 sqft',
 '₹63,045/Month',
 '₹1.1 Crores',
 '1,120 sqft',
 '₹28,657/Month',
 '₹50 Lacs',
 '2,350 sqft',
 '₹71,643/Month',
 '₹1.25 Crores',
 '2,200 sqft',
 '₹56,168/Month',
 '₹98 Lacs']
```

In [129]:

```
del price[0:30:3]
```

In [130]:

```
price
```

Out[130]:

```
['₹45,851/Month',
 '₹80 Lacs',
 '₹85,971/Month',
 '₹1.5 Crores',
 '₹40,120/Month',
 '₹70 Lacs',
 '₹20,060/Month',
 '₹35 Lacs',
 '₹1.43 Lacs/Month',
 '₹2.5 Crores',
 '₹51,583/Month',
 '₹90 Lacs',
 '₹63,045/Month',
 '₹1.1 Crores',
 '₹28,657/Month',
 '₹50 Lacs',
 '₹71,643/Month',
 '₹1.25 Crores',
 '₹56,168/Month',
 '₹98 Lacs']
```

In [131]:

```
del price[0:20:2]
```

In [132]:

```
price
```

Out[132]:

```
['₹80 Lacs',
 '₹1.5 Crores',
 '₹70 Lacs',
 '₹35 Lacs',
 '₹2.5 Crores',
 '₹90 Lacs',
 '₹1.1 Crores',
 '₹50 Lacs',
 '₹1.25 Crores',
 '₹98 Lacs']
```

In [133]:

```
print(len(flat),len(location),len(area),len(emi),len(price))
```

```
10 10 10 10 10
```

In [134]:

```
df = pd.DataFrame()
df
```

Out[134]:

In [135]:

```
df['House Title']=flat
df['Location']=location
df['Area (sqft)']=area
df['EMI / Month']=emi
df['Price']=price
```

In [136]:

df

Out[136]:

	House Title	Location	Area (sqft)	EMI / Month	Price
0	For Sale In Naganathapura Independent House, ...	Independent House, Doddanagamangala Rd opposit...	1,50045,851/MonthEstimated EMI80 Lacs5,333 pe...	1,500 sqftBuiltup₹45,851 EMI₹80 Lacs₹5,333 pe...	₹80 Lacs
1	For Sale In Daadys Garden In Electronic City ...	Daadys Garden Kammasandra Rd, Kammasandra, El...	2,600	₹85,971	₹1.5 Crores
2	For Sale In Sarjapura Independent House, Sha...	Independent House, Shantipura Village , S.P L...	1,100	₹40,120	₹70 Lacs
3	Flat For Sale In Electronic City Standalone ...	Standalone Building, Shikaripalya, near Shams...	1,400	₹20,060	₹35 Lacs

In []:

In []:

In []: