```
import requests

from bs4 import BeautifulSoup
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize
import nltk


page = requests.get("https://www.rjcollege.edu.in/about-us/")
sout = BeautifulSoup(page.content,'html.parser')
str3 = sout.find_all('p')[2].get_text()


str3
```

```
'Shri Nandikishore Singh Jairamji'
```

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
sents = sent_tokenize(str3)


sents
```

```
['Shri Nandikishore Singh Jairamji']
```

```
words = word_tokenize(str3)


words
```

```
['Shri', 'Nandikishore', 'Singh', 'Jairamji']
```

```
sout.title
```

```
<title>ABOUT US – R J COLLEGE</title>
```

```
sout.title.name
```

```
'title'
```

```
sout.title.parent.name
```

```
'head'
```

```
# https://medium.com/analytics-vidhya/how-to-scrape-a-table-from-website-using-python-ce90

fees = requests.get("https://www.rjcollege.edu.in/pgadmission2022-23/")

fees_sout = BeautifulSoup(fees.content,'html.parser')
table1 = fees_sout.find('table',id='tablepress-56')

table1
```

```
<table class="tablepress tablepress-id-56" id="tablepress-56">
<thead>
<tr class="row-1 odd">
<th class="column-1">Course</th><th class="column-2">Eligibility</th><th class="colum
</tr>
</thead>
<tbody class="row-hover">
<tr class="row-2 even">
<td class="column-1">M.Sc Chemistry (Physical, Organic &amp;amp; Inorganic)</td><td class
</tr>
<tr class="row-3 odd">
<td class="column-1">M.Sc Chemistry (Analytical)</td><td class="column-2">B.Sc Chemis
</tr>
<tr class="row-4 even">
<td class="column-1">M.Sc Botany</td><td class="column-2">B.Sc Botany</td><td class='
</tr>
<tr class="row-5 odd">
<td class="column-1">M.Sc Zoology</td><td class="column-2">B.Sc Zoology</td><td class
</tr>
<tr class="row-6 even">
<td class="column-1">M.Sc Physics</td><td class="column-2">B.Sc Physics</td><td class
</tr>
<tr class="row-7 odd">
<td class="column-1">M.Sc Biotechnology</td><td class="column-2">B.Sc Biotechnology</
</tr>
<tr class="row-8 even">
<td class="column-1">M.Sc Computer Science</td><td class="column-2">B.Sc I.T/CS/Stati
</td><td class="column-3">30 June 2022 at 6.00 PM</td><td class="column-4">1July 2022
</tr>
<tr class="row-9 odd">
<td class="column-1">M.Sc Information Technology</td><td class="column-2">B.Sc I.T/CS
</tr>
<tr class="row-10 even">
<td class="column-1">M.Sc Data  Science &amp;amp; Artificial Intelligence (DSAI)</td><td
</tr>
<tr class="row-11 odd">
<td class="column-1">M.Sc M.Sc Environmental Science &amp;amp; Disaster Management (ESDM)
</tr>
<tr class="row-12 even">
<td class="column-1">M.Sc Statistics</td><td class="column-2">B.Sc Statistics</td><td
</tr>
<tr class="row-13 odd">
<td class="column-1">M.A Hindi</td><td class="column-2">Any Graduate (for students ot
</tr>
<tr class="row-14 even">
<td class="column-1">M.A English</td><td class="column-2">Any Graduate (for students
</tr>
<tr class="row-15 odd">
```

```
<td class="column-1">M.A EMA</td><td class="column-2">Any Graduate</td><td class="col
</tr>
<tr class="row-16 even">
<td class="column-1">M..Com - Advanced Accountancy and M.Com - Business management</t
</tr>
</tbody>
</table>
```

```
headers = []
for i in table1.find_all('th'):
 title = i.text
 headers.append(title)
```

```
headers
```

```
['Course',
 'Eligibility',
 'Last date of Online form Filling',
 'Display of Provisional List',
 'Counseling and allotment of Seats on the basis of Merit',
 'Admission Criteria']
```

```
import pandas as pd
mydata = pd.DataFrame(columns = headers)
```

```
mydata
```

Course	Eligibility	Last date of Online form Filling	Display of Provisional List	Counseling and allotment of Seats on the basis of Merit	Admission Criteria

```
for j in table1.find_all('tr')[1:]:
 row_data = j.find_all('td')
 row = [i.text for i in row_data]
 length = len(mydata)
 mydata.loc[length] = row
```

```
mydata
```

	Course	Eligibility	Last date of Online form Filling	Display of Provisional List	Counseling and allotment of Seats on the basis of Merit	Admission Criteria
0	M.Sc Chemistry (Physical, Organic & Inorganic)	B.Sc Chemistry	30 June 2022 at 6.00 PM	1July 2022 at 6.00 PM	Will be announced soon	on the Basis of Merit Counselling will be done...
1	M.Sc Chemistry (Analytical)	B.Sc Chemistry	30 June 2022 at 6.00 PM	1July 2022 at 6.00 PM	Will be announced soon	on the Basis of Merit Counselling will be done...
2	M.Sc Botany	B.Sc Botany	30 June 2022 at 6.00 PM	1July 2022 at 6.00 PM	Will be announced soon	on the Basis of Merit Counselling will be done...
3	M.Sc Zoology	B.Sc Zoology	30 June 2022 at 6.00 PM	1July 2022 at 6.00 PM	Will be announced soon	on the Basis of Merit Counselling will be

```
gallery_img = requests.get("https://www.rjcollege.edu.in/gallery/")

gallery_sout = BeautifulSoup(gallery_img.content, 'html.parser')

gallery_img_lst = []

imgs = gallery_sout.select('img')

for img in imgs:
  s = img.get('src')
  a = img.get('alt')

  gallery_img_lst.append({"src": s,"alt": a})

for img in gallery_img_lst:
  print(img)
```

```
{'src': '
{'src': 'https://www.rjcollege.edu.in/wp-content/uploads/2021/12/website-logo-1.png
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
```

```
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
{'src': 'data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F2000%2Fsv
```


```
xml_eg = requests.get("https://www.w3schools.com/xml/note.xml")
```

```
xml_sout = BeautifulSoup(xml_eg.content, 'xml')
```

```
xml_sout
```

```
<?xml version="1.0" encoding="utf-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

```
x = xml_sout.find('from')
x
```

```
<from>Jani</from>
```

```
# https://medium.com/@cmukesh8688/web-scraping-json-dictionary-and-pandas-part-2-4d3443228
```

```
json_eg = requests.get('https://maps2.dcgis.dc.gov/dcgis/rest/services/FEEDS/MPD/MapServer
```

```
json_eg.status_code
```

```
200
```

```
type(json_eg)
```

```
requests.models.Response
```

```
d_json_eg = json_eg.json()
d_json_eg
```


```
{'displayFieldName': 'CCN',
 'exceededTransferLimit': True,
 'features': [{'attributes': {'ANC': '8A',
    'BID': 'ANACOSTIA',
    'BLOCK': '1200 - 1299 BLOCK OF GOOD HOPE ROAD SE',
    'BLOCK_GROUP': '007503 1',
    'CCN': '20040945',
    'CENSUS_TRACT': '007503',
    'DISTRICT': '7',
    'END_DATE': 1583602253000,
    'LATITUDE': 38.8672656498,
    'LONGITUDE': -76.9878419796,
    'METHOD': 'OTHERS',
    'NEIGHBORHOOD_CLUSTER': 'Cluster 28',
    'OBJECTID': 175253942,
    'OCTO_RECORD_ID': None,
    'OFFENSE': 'THEFT/OTHER',
    'PSA': '701',
    'REPORT_DAT': 1583609902000,
    'SHIFT': 'DAY',
    'START_DATE': 1583596843000,
    'VOTING_PRECINCT': 'Precinct 114',
```

```
    'WARD': '8',
    'XBLOCK': 401055.1176309321,
    'YBLOCK': 133271.6069371067},
   'geometry': {'x': -76.98784425767002, 'y': 38.867273431546366}},
  {'attributes': {'ANC': '5E',
    'BID': None,
    'BLOCK': '1 - 99 BLOCK OF Q STREET NW',
    'BLOCK_GROUP': '004600 1',
    'CCN': '20040957',
    'CENSUS_TRACT': '004600',
    'DISTRICT': '3',
    'END_DATE': 1583604937000,
    'LATITUDE': 38.9111209077,
    'LONGITUDE': -77.0109900409,
    'METHOD': 'OTHERS',
    'NEIGHBORHOOD_CLUSTER': 'Cluster 21',
    'OBJECTID': 175253943,
    'OCTO_RECORD_ID': None,
    'OFFENSE': 'THEFT/OTHER',
    'PSA': '308',
    'REPORT_DAT': 1583616006000,
    'SHIFT': 'EVENING',
    'START_DATE': 1583497820000,
    'VOTING_PRECINCT': 'Precinct 19',
    'WARD': '5',
    'XBLOCK': 399046.83,
    'YBLOCK': 138139.87},
   'geometry': {'x': -77.01099232855701, 'y': 38.911128697745966}},
  {'attributes': {'ANC': '6C',
    'BID': 'NOMA',
    'BLOCK': '1200 - 1229 BLOCK OF 1ST STREET NE',
    'BLOCK_GROUP': '010601 2',
    'CCN': '20040978',
    'CENSUS_TRACT': '010601',
    'DISTRICT': '5',
    'END_DATE': 1583612029000
```


```
for x in d_json_eg:
  print(x)

print("\n")

print(d_json_eg["features"])


df = pd.DataFrame(d_json_eg['features'])
df.head()
```

	attributes	geometry
0	{'CCN': '20040945', 'REPORT_DAT': 158360990200...	{'x': -76.98784425767002, 'y': 38.867273431546...
1	{'CCN': '20040957', 'REPORT_DAT': 158361600600...	{'x': -77.01099232855701, 'y': 38.911128697745...
2	{'CCN': '20040978', 'REPORT_DAT': 158361453900...	{'x': -77.00588677394491, 'y': 38.90605178244354}
3	{'CCN': '20040993', 'REPORT_DAT':	{'x': -76.9984525859422, 'y':