```
1.
import requests
from bs4 import BeautifulSoup

url = "https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos"

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

table = soup.find("table", class_="wikitable")

rank_list = []
name_list = []
artist_list = []
upload_date_list = []
views_list = []

for row in table.find_all("tr")[1:]:
    columns = row.find_all("td")
    rank = columns[0].text.strip()
    name = columns[1].text.strip()
    artist = columns[2].text.strip()
    upload_date = columns[3].text.strip()
    views = columns[4].text.strip()

    rank_list.append(rank)
    name_list.append(name)
    artist_list.append(artist)
    upload_date_list.append(upload_date)
    views_list.append(views)

for rank, name, artist, upload_date, views in zip(rank_list, name_list, artist_list, upload_date_list,
views_list):
    print("Rank:", rank)
```

```
    print("Name:", name)
    print("Artist:", artist)
    print("Upload Date:", upload_date)
    print("Views:", views)
    print("-----------------------------")
```

2.

```
import requests
from bs4 import BeautifulSoup

url = "https://www.bcci.tv/"

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

fixtures_section = soup.find("section", class_="latest-series")

fixtures = fixtures_section.find_all("div", class_="fixture__info")

series_list = []
place_list = []
date_list = []
time_list = []

for fixture in fixtures:
    series = fixture.find("p", class_="fixture__additional-info").text.strip()
    place = fixture.find("span", class_="fixture__location").text.strip()
    date = fixture.find("span", class_="fixture__datetime").text.strip().split(", ")[0]
    time = fixture.find("span", class_="fixture__datetime").text.strip().split(", ")[1]

    series_list.append(series)
    place_list.append(place)
    date_list.append(date)
```

```
    time_list.append(time)


for series, place, date, time in zip(series_list, place_list, date_list, time_list):
    print("Series:", series)
    print("Place:", place)
    print("Date:", date)
    print("Time:", time)
    print("-----------------------------")
```

3.

```
import requests
from bs4 import BeautifulSoup

url = "http://statisticstimes.com/"

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

state_gdp_link = soup.find("a", href="/economy/gdp-of-indian-states.php")

if state_gdp_link:
    state_gdp_url = "http://statisticstimes.com" + state_gdp_link["href"]

    state_gdp_response = requests.get(state_gdp_url)

    state_gdp_soup = BeautifulSoup(state_gdp_response.text, "html.parser")

    table = state_gdp_soup.find("table", class_="display")

    rank_list = []
    state_list = []
```

```
    gdp_18_19_list = []
    gdp_19_20_list = []
    share_18_19_list = []
    gdp_billion_list = []

    for row in table.find_all("tr")[1:]:
        columns = row.find_all("td")
        rank = columns[0].text.strip()
        state = columns[1].text.strip()
        gdp_18_19 = columns[2].text.strip()
        gdp_19_20 = columns[3].text.strip()
        share_18_19 = columns[4].text.strip()
        gdp_billion = columns[5].text.strip()

        rank_list.append(rank)
        state_list.append(state)
        gdp_18_19_list.append(gdp_18_19)
        gdp_19_20_list.append(gdp_19_20)
        share_18_19_list.append(share_18_19)
        gdp_billion_list.append(gdp_billion)



    for rank, state, gdp_18_19, gdp_19_20, share_18_19, gdp_billion in zip(
        rank_list, state_list, gdp_18_19_list, gdp_19_20_list, share_18_19_list, gdp_billion_list
    ):
        print("Rank:", rank)
        print("State:", state)
        print("GSDP(18-19) - at current prices:", gdp_18_19)
        print("GSDP(19-20) - at current prices:", gdp_19_20)
        print("Share(18-19):", share_18_19)
        print("GDP($ billion):", gdp_billion)
        print("-----------------------------")
else:
    print("State-wise GDP data link not found on the website.")
```

4.

```
import requests
from bs4 import BeautifulSoup

url = "https://github.com/"

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

trending_section = soup.find("div", class_="explore-pjax-container container-lg p-responsive pt-6")

repositories = trending_section.find_all("article", class_="Box-row")

titles = []
descriptions = []
contributors_counts = []
languages_used = []

for repository in repositories:
    title = repository.find("h1", class_="h3 lh-condensed").text.strip()
    titles.append(title)

    description = repository.find("p", class_="col-9 text-gray my-1 pr-4").text.strip()
    descriptions.append(description)

    contributor_count = repository.find("a", class_="Link--muted d-inline-block mr-3").text.strip()
    contributors_counts.append(contributor_count)

    language_used = repository.find("span", itemprop="programmingLanguage")
    if language_used:
        language_used = language_used.text.strip()
```

```
    else:
        language_used = "Not specified"
    languages_used.append(language_used)

for i in range(len(titles)):
    print("Repository Title:", titles[i])
    print("Repository Description:", descriptions[i])
    print("Contributors Count:", contributors_counts[i])
    print("Language Used:", languages_used[i])
    print("-----------------------------")
```

5.

```
import requests
from bs4 import BeautifulSoup

url = "https://www.billboard.com/"

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

top_100_section = soup.find("section", class_="top-100")

songs = top_100_section.find_all("div", class_="song")

song_names = []
artist_names = []
last_week_ranks = []
peak_ranks = []
weeks_on_board = []

for song in songs:
    song_name = song.find("span", class_="chart-element__information__song").text.strip()
```

```
    artist_name = song.find("span", class_="chart-element__information__artist").text.strip()
    last_week_rank = song.find("span", class_="chart-element__meta text--center color--secondary text--last").text.strip()
    peak_rank = song.find("span", class_="chart-element__meta text--center color--secondary text--peak").text.strip()
    weeks_on_board = song.find("span", class_="chart-element__meta text--center color--secondary text--week").text.strip()

    song_names.append(song_name)
    artist_names.append(artist_name)
    last_week_ranks.append(last_week_rank)
    peak_ranks.append(peak_rank)
    weeks_on_board.append(weeks_on_board)

for i in range(len(song_names)):
    print("Song Name:", song_names[i])
    print("Artist Name:", artist_names[i])
    print("Last Week Rank:", last_week_ranks[i])
    print("Peak Rank:", peak_ranks[i])
    print("Weeks on Board:", weeks_on_board[i])
    print("---------------“)
```

6.

```
import requests
from bs4 import BeautifulSoup

url = "https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey-compare"

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")
```

```
table = soup.find("table", class_="in-article sortable")

book_names = []
author_names = []
volumes_sold = []
publishers = []
genres = []

for row in table.find_all("tr")[1:]:
    columns = row.find_all("td")
    book_name = columns[0].text.strip()
    author_name = columns[1].text.strip()
    volumes = columns[2].text.strip()
    publisher = columns[3].text.strip()
    genre = columns[4].text.strip()

    book_names.append(book_name)
    author_names.append(author_name)
    volumes_sold.append(volumes)
    publishers.append(publisher)
    genres.append(genre)

for i in range(len(book_names)):
    print("Book Name:", book_names[i])
    print("Author Name:", author_names[i])
    print("Volumes Sold:", volumes_sold[i])
    print("Publisher:", publishers[i])
    print("Genre:", genres[i])
    print("-----------------------------")
```

7.

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

```
url = "https://www.imdb.com/list/ls095964455/"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

tv_series_items = soup.find_all("div", class_="lister-item-content")

for item in tv_series_items:
    name = item.find("h3").a.text.strip()

    year_span = item.find("span", class_="lister-item-year").text.strip()

    genre = item.find("span", class_="genre").text.strip()

    runtime = item.find("span", class_="runtime").text.strip()

    ratings = item.find("div", class_="ipl-rating-star").span.text.strip()

    votes = item.find("span", {"name": "nv"})["data-value"]

    print("Name:", name)
    print("Year Span:", year_span)
    print("Genre:", genre)
    print("Run Time:", runtime)
    print("Ratings:", ratings)
    print("Votes:", votes)
    print("\n")
```