PRACTICAL NO. 5

AIM: Explain web scraping using beautifulSoup, html parser, find_all and dataframes.

THEORY:

Web scraping is the process of extracting data from websites by parsing the HTML code of web pages. It can be a very powerful tool for data collection, analysis and automation.

BeautifulSoup is a Python library that is commonly used for web scraping. It is an HTML and XML parser that makes it easy to extract data from web pages. BeautifulSoup provides a simple way to navigate the HTML tree structure and extract the data we need.

HTML parser is a built-in Python module that is used to parse HTML documents. It provides a simple interface for parsing and manipulating HTML documents. In combination with BeautifulSoup, it is very powerful for web scraping.

find_all() is a BeautifulSoup method that is used to extract all elements from the HTML that match a specific tag or set of tags. It returns a list of all the matching elements, and we can then extract the data we need from those elements.

Dataframes are a data structure in Python that is commonly used for data manipulation and analysis. Data can be organized in rows and columns, and we can perform various operations on the data using libraries like Pandas.

Here is an example of how to use BeautifulSoup, HTML parser, find_all() and dataframes for web scraping:

Code:

```
import requests

from bs4 import BeautifulSoup
```

```
import pandas as pd

# send a request to the webpage
url = "https://www.example.com"
response = requests.get(url)

# parse the HTML content
soup = BeautifulSoup(response.content, 'html.parser')

# find all the links on the webpage
links = soup.find_all('a')

# extract the href attribute from each link and store it in a list
hrefs = []
for link in links:
    href = link.get('href')
    hrefs.append(href)

# create a dataframe from the list of hrefs
df = pd.DataFrame(hrefs, columns=['hrefs'])

# print the first 10 rows of the dataframe
print(df.head(10))
```

CONCLUSION:

Thus we have successfully executed programs .