13/04/24

Today’s work

Today at 09.00 we started practicing python for strong basic knowledge

Web Scrapping

Web scraping is a general term for techniques involving automating the gathering of data from a website.

\* things we need to understand

Rules of web Scraping

Limitations of web scaping

Basic html and css

\*Rules of web scraping

Always try to get permission before scrapping.

If you make too many scraping attempts or requests your IP address could get blocked!

Some sites automatically block scraping software.

* Limitation of web Scraping

In general every website is unique, that means every web scraping script is unique.

A slight change or update to a website may completely break your web scraping script.

Html is used to create the basic structure and content of a webpage.

CSS is used for the design and style of a web page, where elements are placed and how it looks

JavaScript is used to define the interactive elements of a webpage.

To web scrape with python we can use the beautifulSoup and requests libraries.

These are external libraries so we need to install them with pip at your command line.

Directly at your command line use:

Pip install requests

Pip install lxml

Pip install bs4

The requests library in python is a popular http library for making http requests simpler and more human-friendly. It provides methods to send HTTP requests such as GET, POST, PUT, DELETE, etc., and handle the responses easily.

Lxml is a Python library that provides a fast and flexible way to work with XML and HTML documents. It is built on top of the libxml2 and libxslt libraries, which are written in C and known for their speed and efficiency.

Bs4 library in Python, also known as Beautiful Soup 4, is a powerful library for parsing HTML and XML documents. It provides a convenient way to extract and manipulate data from HTML and XML files, making web scraping tasks easier.

import requests

import bs4

result = requests.get("http://www.example.com")

print(type(result))

print(result.text)

output:

<class 'requests.models.Response'>

<!doctype html>

<html>

<head>

<title>Example Domain</title>

<meta charset="utf-8" />

<meta http-equiv="Content-type" content="text/html; charset=utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<style type="text/css">

body {

background-color: #f0f0f2;

margin: 0;

padding: 0;

font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

}

div {

width: 600px;

margin: 5em auto;

padding: 2em;

background-color: #fdfdff;

border-radius: 0.5em;

box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);

}

a:link, a:visited {

color: #38488f;

text-decoration: none;

}

@media (max-width: 700px) {

div {

margin: 0 auto;

width: auto;

}

}

</style>

</head>

<body>

<div>

<h1>Example Domain</h1>

<p>This domain is for use in illustrative examples in documents. You may use this

domain in literature without prior coordination or asking for permission.</p>

<p><a href="https://www.iana.org/domains/example">More information...</a></p>

</div>

</body>

</html>

import requests

import bs4

result = requests.get("http://www.example.com")

soup = bs4.BeautifulSoup(result.text,"lxml")

print(soup)

output:

<!DOCTYPE html>

<html>

<head>

<title>Example Domain</title>

<meta charset="utf-8"/>

<meta content="text/html; charset=utf-8" http-equiv="Content-type"/>

<meta content="width=device-width, initial-scale=1" name="viewport"/>

<style type="text/css">

body {

background-color: #f0f0f2;

margin: 0;

padding: 0;

font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

}

div {

width: 600px;

margin: 5em auto;

padding: 2em;

background-color: #fdfdff;

border-radius: 0.5em;

box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);

}

a:link, a:visited {

color: #38488f;

text-decoration: none;

}

@media (max-width: 700px) {

div {

margin: 0 auto;

width: auto;

}

}

</style>

</head>

<body>

<div>

<h1>Example Domain</h1>

<p>This domain is for use in illustrative examples in documents. You may use this

domain in literature without prior coordination or asking for permission.</p>

<p><a href="https://www.iana.org/domains/example">More information...</a></p>

</div>

</body>

</html>

import requests

import bs4

result = requests.get("http://www.example.com")

soup = bs4.BeautifulSoup(result.text,"lxml")

print(soup.select("title"))

output: [<title>Example Domain</title>]

around 1 took break than start again around 2

web crawler is used to open all the links in a websites and collect the database of all the links.

Web scraping is used to collect all useful information from the webpage.

Html parsing is a process where we extract text from html code.

Prettify- when we extract html code from a website by using get request. Then the extracted code has not in proper structured. Prettify is used for covert code in tree like condition.

import requests

import bs4

result = requests.get("http://www.example.com")

print(result.status\_code)

output:200 # means we can easily check the website.

HTTP status codes are used to indicate the result of a client's request to the server. They are three-digit numbers that are grouped into different classes, each representing a different type of response.

2\*\* Success: the request was successful.

200 : The request was successful.

201 : The request has been fulfilled, resulting in the creation of a new resource.

204 : The server successfully processed the request but there is no content to send in the response.

3\*\* Redirection

301: The requested resource has been permanently moved to a new location.

302: The requested resources temporarily resides under a different URL.

304: NOT modifield

4\*\* Client error

400: The request connot process the request due to a client error.

401: The request has not been applied because it lack valid authentication credential for the target resource.

403 forbidden: The server understood the request but refuses to authorize it.

404 Not found: The server cannot find the requested resource.

5\*\* Server Error: The server Failed to fulfil a valid request.

500 Internal server error: A generic error message indicating that an unexpected condition was encountered.

503 Server Unavailable: The server is currently unable to handle the request due to a temporary overload or maintenance.

Kind of objects beautifull soup in web scrapping

Tag

Navigablestring

BeautifulSoup

Comments

tag = soup.html

print(tag)

print(type(tag))

print(soup.a)

print(soup.h1)

Navigablestring is used to extract the string from the tags

#print(soup.p.string)

#print(soup.a.string)

#print(soup.h1.string)

Beautifulsoup is used to directly extract the url data

#beautifulsoup

#print(soup.body)

#print(soup.head)

#print(soup.find("p"))

#print(soup.find\_all("p"))

Finding elements

Finding elements by class

Finding elements by id

I got error so that why I wind all this around 6 and upload it on git hub

Total sitting over the day = 2

Total study time over the day = approx.. 8hr