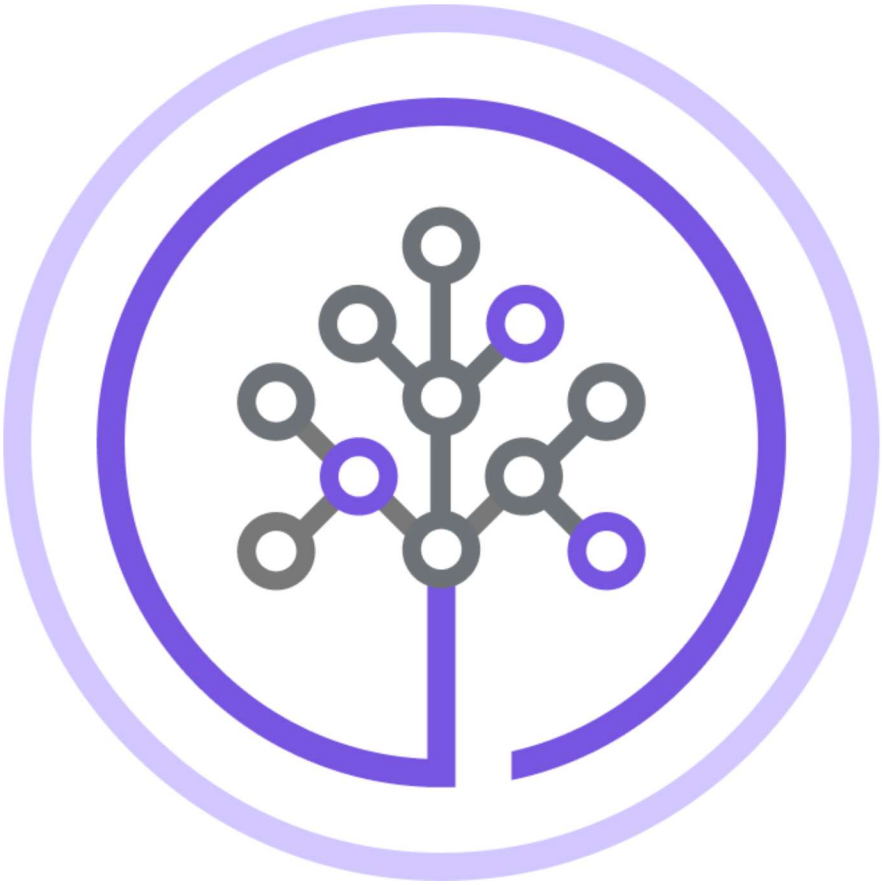


Quét web và HTML cơ bản

Thời gian dự kiến: 10 phút

Mục tiêu

Sau khi hoàn thành việc đọc này, bạn sẽ có thể:

- Giải thích các khái niệm chính liên quan đến cấu trúc HTML và thành phần thẻ HTML.
- Khám phá khái niệm về cây tài liệu HTML.
- Làm quen với các bảng HTML.
- Hiểu rõ hơn về những kiến thức cơ bản về quét web bằng Python và BeautifulSoup.

Giới thiệu về quét web

Quét web, còn được gọi là thu thập web hoặc trích xuất dữ liệu web, là quá trình trích xuất thông tin từ các trang web hoặc trang web. Nó liên quan đến việc truy xuất dữ liệu tự động từ các nguồn web. Mọi người sử dụng nó cho các ứng dụng khác nhau như phân tích dữ liệu, khai thác, so sánh giá, tổng hợp nội dung, v.v.

Cách quét web hoạt động

Yêu cầu HTTP

Quá trình này thường bắt đầu bằng một yêu cầu HTTP. Trình quét web gửi yêu cầu HTTP đến một URL cụ thể, tương tự như cách trình duyệt web thực hiện khi bạn truy cập một trang web. Yêu cầu thường là yêu cầu HTTP GET để truy xuất nội dung của trang web.

Truy xuất trang web

Máy chủ web lưu trữ trang web sẽ phản hồi yêu cầu bằng cách trả về nội dung HTML của trang web được yêu cầu. Nội dung này bao gồm các phần tử văn bản và phương tiện hiển thị cũng như cấu trúc HTML cơ bản xác định bố cục của trang.

Phân tích cú pháp HTML

Sau khi nhận được nội dung HTML, bạn cần phân tích nội dung. Phân tích cú pháp bao gồm việc chia cấu trúc HTML thành các thành phần, chẳng hạn như thẻ, thuộc tính và nội dung văn bản. Bạn có thể sử dụng BeautifulSoup bằng Python. Nó tạo ra một biểu diễn có cấu trúc của nội dung HTML có thể dễ dàng điều hướng và thao tác.

Trích xuất dữ liệu

Với nội dung HTML được phân tích cú pháp, giờ đây người quét web có thể xác định và trích xuất dữ liệu cụ thể mà họ cần. Dữ liệu này có thể bao gồm văn bản, liên kết, hình ảnh, bảng biểu, giá sản phẩm, tin tức, v.v. Người dọn dẹp định vị dữ liệu bằng cách tìm kiếm các thẻ, thuộc tính và mẫu HTML có liên quan trong cấu trúc HTML.

Chuyển đổi dữ liệu

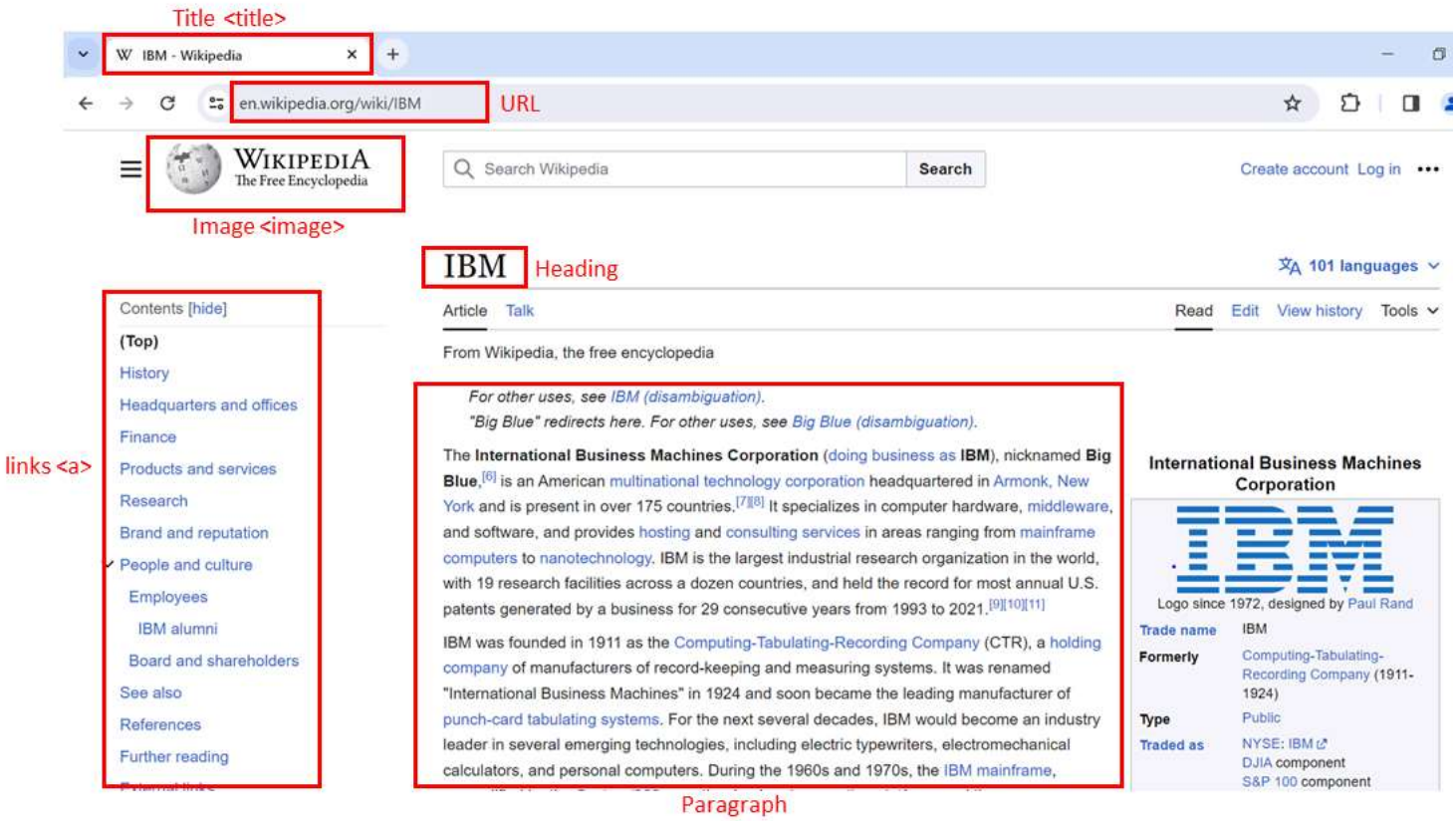
Dữ liệu được trích xuất có thể cần được xử lý và chuyển đổi thêm. Ví dụ: bạn có thể xóa thẻ HTML khỏi văn bản, chuyển đổi định dạng dữ liệu hoặc dọn sạch dữ liệu lộn xộn. Bước này đảm bảo dữ liệu sẵn sàng để phân tích hoặc các trường hợp sử dụng khác.

Kho

Sau khi trích xuất và chuyển đổi, bạn có thể lưu trữ dữ liệu đã được thu thập ở nhiều định dạng khác nhau, chẳng hạn như cơ sở dữ liệu, bảng tính, tệp JSON hoặc CSV. Việc lựa chọn định dạng lưu trữ phụ thuộc vào yêu cầu cụ thể của dự án.

Tự động hóa

Trong nhiều trường hợp, các tập lệnh hoặc chương trình sẽ tự động quét web. Các công cụ tự động hóa này cho phép trích xuất dữ liệu định kỳ từ nhiều trang web hoặc trang web. Quét tự động đặc biệt hữu ích để thu thập dữ liệu từ các trang web động thường xuyên cập nhật nội dung của chúng.



Cấu trúc HTML

Ngôn ngữ đánh dấu siêu văn bản (HTML) đóng vai trò là nền tảng của các trang web. Hiểu cấu trúc của nó là rất quan trọng cho việc quét web.

- `<html>`là phần tử gốc của trang HTML.
- `<head>`chứa thông tin meta về trang HTML.
- `<body>`hiển thị nội dung trên trang web, thường là các dữ liệu quan tâm.
- `<h3>` tags are type 3 headings, making text larger and bold, typically used for player names.
- `<p>` tags represent paragraphs and contain player salary information.

Composition of an HTML tag

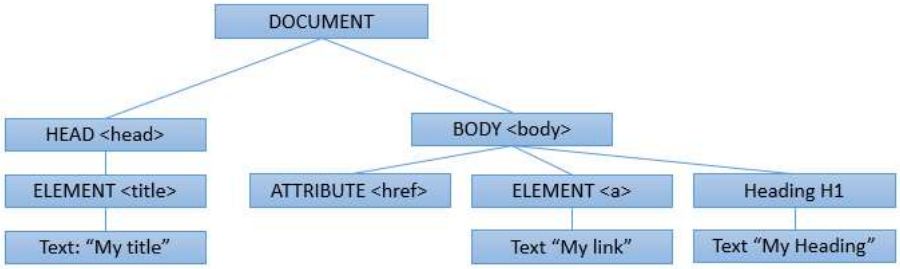
HTML tags define the structure of web content and can contain attributes.

- An HTML tag consists of an opening (start) tag and a closing (end) tag.
- Tags have names (`<a>` for an anchor tag).
- Tags may contain attributes with an attribute name and value, providing additional information to the tag.

HTML document tree

You can visualize HTML documents as trees with tags as nodes.

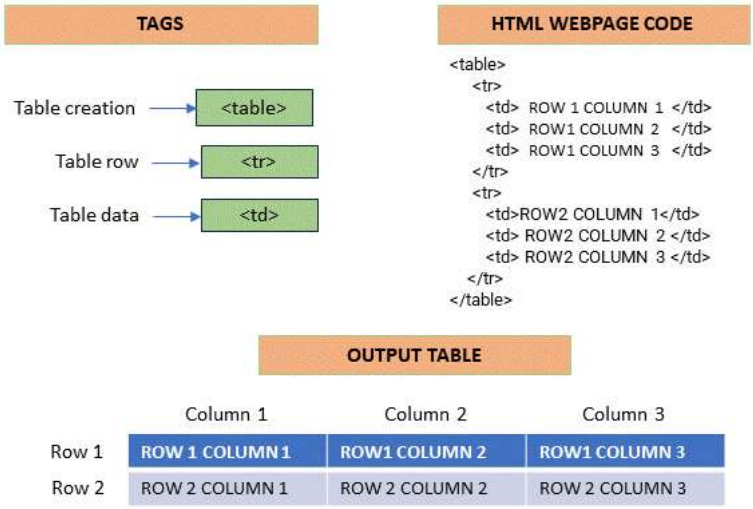
- Tags can contain strings and other tags, making them the tag's children.
- Tags within the same parent tag are considered siblings.
- For example, the `<html>` tag contains both `<head>` and `<body>` tags, making them descendants of `<html` but children of `<html>`. `<head>` and `<body>` are siblings.



HTML tables

HTML tables are essential for presenting structured data.

- Define an HTML table using the `<table>` tag.
- Each table row is defined with a `<tr>` tag.
- The first row often uses the table header tag, typically `<th>`.
- The table cell is represented by `<td>` tags, defining individual cells in a row.



Web scraping

Web scraping involves extracting information from web pages using Python. It can save time and automate data collection.

Required tools

Web scraping requires Python code and two essential modules: Requests and Beautiful Soup. Ensure you have both modules installed in your Python environment.

```
1
2
```

```
# Import Beautiful Soup to parse the web page content
from bs4 import BeautifulSoup
```

Copied!

Fetching and parsing HTML

To start web scraping, you need to fetch the HTML content of a webpage and parse it using Beautiful Soup. Here's a step-by-step example:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

```
16. 16
17. 17
```

```
 1. import requests
 2. from bs4 import BeautifulSoup
 3.
 4. # Specify the URL of the webpage you want to scrape
 5. url = 'https://en.wikipedia.org/wiki/IBM'
 6.
 7. # Send an HTTP GET request to the webpage
 8. response = requests.get(url)
 9.
10. # Store the HTML content in a variable
11. html_content = response.text
12.
13. # Create a BeautifulSoup object to parse the HTML
14. soup = BeautifulSoup(html_content, 'html.parser')
15.
16. # Display a snippet of the HTML content
17. print(html_content[:500])
```

Copied!

Navigating the HTML structure

BeautifulSoup represents HTML content as a tree-like structure, allowing for easy navigation. You can use methods like find_all to filter and extract specific HTML elements. For example, to find all anchor tags () and print their text:

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
```

```
1. # Find all <a> tags (anchor tags) in the HTML
2. links = soup.find_all('a')
3.
4. # Iterate through the list of links and print their text
5. for the link in links:
6.     print(link.text)
```

Copied!

Custom data extraction

Web scraping allows you to navigate the HTML structure and extract specific information based on your requirements. This process may involve finding specific tags, attributes, or text content within the HTML document.

Using BeautifulSoup for HTML parsing

Beautiful Soup is a powerful tool for navigating and extracting specific web page parts. It allows you to find elements based on their tags, attributes, or text, making extracting the information you're interested in easier.

Using pandas read_html for table extraction

Pandas, a Python library, provides a function called read_html, which can automatically extract data from websites' tables and present it in a format suitable for analysis. It's similar to taking a table from a webpage and importing it into a spreadsheet for further analysis.

Conclusion

Trong bài đọc này, bạn đã tìm hiểu về quét web bằng BeautifulSoup và Pandas, nhấn mạnh vào việc trích xuất các phần tử và bảng. BeautifulSoup hỗ trợ phân tích cú pháp HTML, trong khi read_html của Pandas đơn giản hóa việc trích xuất bảng. Bài đọc cũng nhấn mạnh việc quét web có trách nhiệm, đảm bảo tuân thủ các điều khoản của trang web. Được trang bị kiến thức này, bạn có thể tự tin tham gia vào việc trích xuất dữ liệu chính xác.

Tác giả

Akansha Yadav