WEB SCRAPING

domenica 22 ottobre 2023 11:56

Web Scraping:

Il web scraping è il processo di estrazione di dati da un sito web e della loro conversione in un formato strutturato. Questo può essere utilizzato in vari modi, tra cui il confronto dei prezzi, la raccolta di indirizzi email, lo scraping di contenuti dai social media, la ricerca e lo sviluppo di informazioni, e la raccolta di annunci di lavoro.

Legalità del Web Scraping:

La maggior parte dei dati visualizzati sui siti web è generalmente accessibile al pubblico, il che significa che l'estrazione di tali dati non è di per sé illegale. Tuttavia, è importante rispettare i termini di servizio e il file `robots.txt` di un sito web, se presenti, per garantire che il web scraping sia condotto in modo etico e conforme alle regole del sito.

Robots.txt:

Il file `robots.txt` è un file di testo creato per istruire i robot web, in particolare i robot dei motori di ricerca, su come eseguire la scansione delle pagine di un sito web. Le istruzioni all'interno di `robots.txt` possono specificare se un particolare comportamento è "consentito" o "non consentito" per determinati agenti utenti.

La sintassi di `robots.txt` comprende:

- `User-agent:`: Specifica l'agente utente (spesso un motore di ricerca) a cui si stanno fornendo istruzioni di scansione.

- `Disallow:`: Indica agli agenti utenti di non eseguire la scansione di URL specifici, limitando l'accesso a parti del sito.

- `Allow` (applicabile solo a Googlebot): Comunica a Googlebot che può accedere a pagine o sottocartelle anche se la pagina principale o la sottocartella correlata potrebbe essere contrassegnata come "Disallow."

- `Crawl-delay`: Specifica un ritardo in secondi che i crawler devono attendere prima di scannerizzare una pagina, aiutando a evitare un carico eccessivo sul server web.

- `Sitemap`: Indica la posizione di un file XML della mappa del sito associato all'URL.

Un esempio di file `robots.txt` potrebbe essere:

```
User-agent: Googlebot
Disallow: /private/
Allow: /public/
Crawl-delay: 10
Sitemap: https://www.example.com/sitemap.xml
```

Document Object Model (DOM):

Il Document Object Model (DOM) è un'interfaccia di

programmazione utilizzata per rappresentare e manipolare documenti HTML e XML. Il DOM definisce la struttura logica di questi documenti e fornisce un modo per accedervi e modificarli utilizzando JavaScript.

Il DOM tratta il contenuto testuale e gli elementi del documento web come oggetti che possono essere gestiti da JavaScript. Tuttavia, il DOM non è progettato per valutare la rilevanza degli oggetti nei documenti e non fornisce informazioni sul contesto o sulla semantica degli oggetti nel documento.

Il DOM offre metodi come `write`, `getElementById`, `getElementsByName`, `getElementsByTagName`, e `getElementsByClassName` per accedere e manipolare il contenuto del documento.

Python per il Web Scraping:

Python è ampiamente utilizzato per il web scraping per diversi motivi:

- Facilità d'uso: La programmazione in Python è semplice da scrivere e leggere, grazie alla sua sintassi intuitiva.
- Ampia raccolta di librerie: Python offre una vasta collezione di librerie come Numpy, Matplotlib, e Pandas, che semplificano il web scraping e l'analisi dei dati.
- Duck typing: In Python, non è necessario dichiarare esplicitamente i tipi di dati, rendendo la scrittura di codice più flessibile.
- Sintassi comprensibile: La sintassi di Python è facilmente comprensibile, rendendo il codice leggibile come l'inglese.
- Efficienza: Python è efficace nel gestire compiti di web

- Efficienza: Python è efficace nel gestire compiti di web scraping e può risparmiare tempo nella raccolta di dati.

Selenium:

Selenium è un framework di test automatizzato open source utilizzato per convalidare applicazioni web su diversi browser e piattaforme. Oggi, Selenium viene ampiamente utilizzato per scopi di web scraping e automazione. Può simulare interazioni con il browser, come clic sui pulsanti, compilazione di moduli, scorrimento delle pagine e persino la cattura di screenshot delle pagine web.