



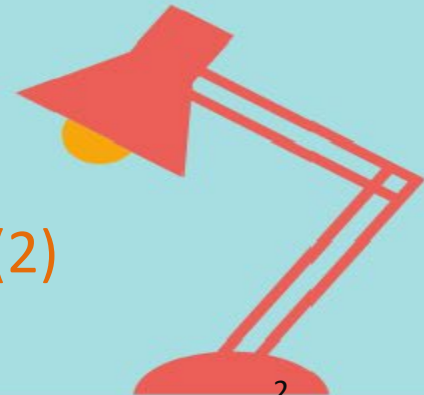
111-2進階程式設計課程(7)

Advanced Computer Programming

亞大資工系

課程大綱

- W1-課程介紹/Introduction
- W2-Python libraries
- W3-BeautifulSoup(1)
- W4-BeautifulSoup(2)
- W5-
- W6-Scrapy(1)
- W7-Scrapy(2)
- W8-Storing Data
- W9-Midterm project
- W10-Web & HTTP
- W11-Flask
- W12-Flask Routes
- W13-Jinja template
- W14-Flask-form
- W15-Flask-mail
- W16-REST API
- W17-Project development(2)
- W18-Final presentation



教材Github

htchu / ACP110Course Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

htchu Init a77a42a 28 seconds ago 2 commits

docs	Init	28 seconds ago
notebooks	Init	28 seconds ago
slides	Init	28 seconds ago
LICENSE	Initial commit	1 hour ago
README.md	Initial commit	1 hour ago

README.md

ACP110Course

亞洲大學110-2進階程式設計課程

大綱

- Review-Web Crawlers
- Review-robots.txt
 - https://developers.google.com/search/docs/advanced/robots/robots_txt
- Review-Regular Expressions
- 作業 2(Assignment 2):



作業1/Assignment 1

- 概述：
 - 在本次作業中，我們將使用基本的 Python 網頁抓取工具 `urllib` 和 `BeautifulSoup` 從 `cna` 網站抓取數據。請列出抓取的新聞內容並將其提交到 Tronclass 的作業條目。
- 目標：
 - 了解如何使用網頁抓取獲取網頁內容。
 - 探索真正的 `html` 文件。
 - 反思網絡抓取功能在數據科學中的可能用途。
- 指示：
 - 使用任何瀏覽器訪問 `www.cna.com.tw` 網站。
 - 檢查 `html` 內容中的標籤。



作業2/Assignment 2

- 概述：
 - 在本次作業中，我們將使用BeautifulSoup從 cna 網站抓取所有的新聞內容並將其存到一個txt檔提交到 Tronclass 的作業條目。
- 目標：
 - 了解如何使用網頁抓取獲取網頁link。
 - 探索crawling an **entire site**。
 - 反思網絡抓取功能在數據科學中的可能用途。
- 指示：
 - 使用任何瀏覽器訪問 focustaiwan 網站。 www.cna.com.tw
 - 檢查 html 內容中的標籤。



期中報告

- 概述：
 - 練習從亞大資工系的網頁
https://csie.asia.edu.tw/zh_tw/associate_professors_2
 - 讀取每個老師的專長 / Discipline expertise
 - 存成一個文字檔，可以是txt, csv或json格式
 - 程式部份可以交notebook或py檔。
- 目標：
 - 了解如何使用網頁抓取獲取網頁link。

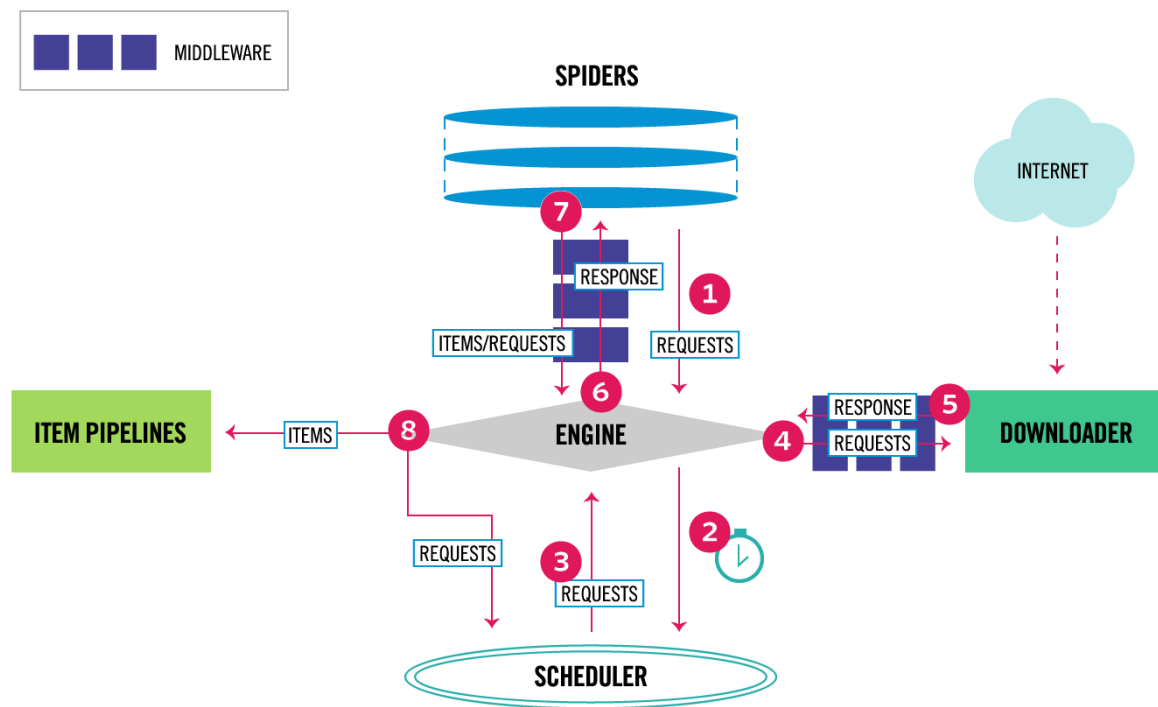


Scrapy爬蟲框架

- **Scrapy Engine(引擎)**: 負責Spider、ItemPipeline、Downloader、Scheduler中間的通訊，信號、數據傳遞等。
- **Scheduler(調度器)**: 它負責接受引擎發送過來的Request請求，並按照一定的方式進行整理排列，入隊，當引擎需要時，交還給引擎。
- **Downloader (下載器)**：負責下載Scrapy Engine(引擎)發送的所有Requests請求，並將其獲取到的Responses交還給Scrapy Engine(引擎)，由引擎交給Spider來處理。
- **Spider (爬蟲)**：它負責處理所有Responses,從中分析提取數據，獲取Item欄位需要的數據，並將需要跟進的URL提交給引擎，再次進入Scheduler(調度器)。
- **Item Pipeline(管道)**：它負責處理Spider中獲取到的Item，並進行進行後期處理（詳細分析、過濾、存儲等）的地方。
- **Downloader Middlewares (下載中間件)**：你可以當作是一個可以自定義擴展下載功能的組件。
- **Spider Middlewares (Spider中間件)**：你可以理解為是一個可以自定擴展和操作引擎和Spider中間通信的功能組件（比如進入Spider的Responses;和從Spider出去的Requests）



Scrapy爬蟲流程



Scrapy Engine(引擎):

Scheduler(調度器):

Downloader (下載器)

Spider (爬蟲)

Item Pipeline(管道)

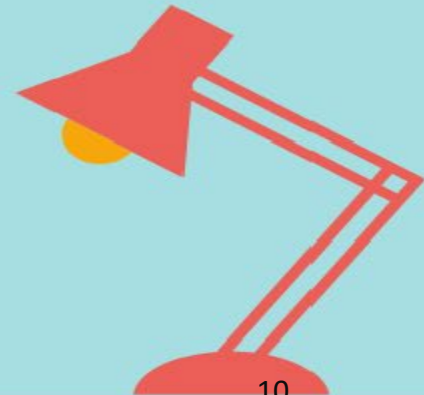
Downloader Middlewares (下載中間件)

Spider Middlewares (Spider中間件)

1. Spider發送最初的請求(Requests)給Engine。
2. Engine在Scheduler調度一個請求(Requests)，並要求下一次Requests做爬取。
3. Scheduler回傳下一個Requests給Engine。
4. Engine透過Downloader Middlewares發送請求給Downloader。
5. 只要頁面結束下載，Downloader產生一個Response透過Downloader Middlewares傳送給Engine。
6. Engine收到來自Downloader的Response並透過Spider Middlewares發送給Spider處理。
7. Spider處理Response並爬取的項目(item)和新的請求(Requests)，透過Spider Middlewares回傳給Engine。
8. Engine發送處理的項目(item)給Item Pipelines接著發送處理的請求(Requests)到Scheduler要求下一個可能的爬蟲請求。

SQLite

- SQLite是小型關聯式數據庫管理系統，它包含在一個相對小的C程式庫中。與許多其它數據庫管理系統不同，SQLite不是一個客戶端/伺服器結構的數據庫引擎，而是被整合在用戶程式中。
- SQLite實現了大多數SQL標準。它使用動態的、弱類型的SQL語法。它作為嵌入式數據庫，是應用程式，如網頁瀏覽器，在本地/客戶端儲存資料的常見選擇。它可能是最廣泛部署的數據庫引擎，因為它正在被一些流行的瀏覽器、作業系統、嵌入式系統所使用。
- SQLite是D. Richard Hipp建立的。



Python sqlite3 -SQLite DB-API 2.0

```
import sqlite3

con = sqlite3.connect('example.db')

cur = con.cursor()

# Create table

cur.execute('CREATE TABLE stocks (date text, trans text, symbol text, qty real, price real)')

# Insert a row of data

cur.execute("INSERT INTO stocks VALUES ('2006-01-05','BUY','RHAT',100,35.14)")

# Save (commit) the changes

con.commit()

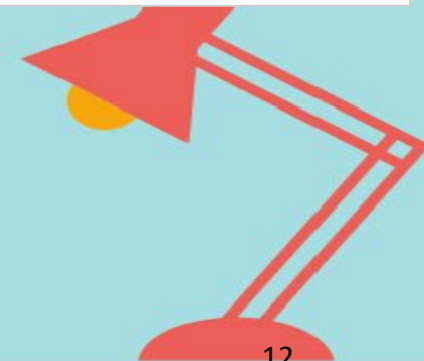
# We can also close the connection if we are done with it.
# Just be sure any changes have been committed or they will be lost.

con.close()
```

Meta Commands

```
系統管理員: Anaconda Prompt (Anaconda3)
(base) D:\Python\crawler>sqlite3
SQLite version 3.36.0 2021-06-18 18:36:39
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .quit
(base) D:\Python\crawler>
```

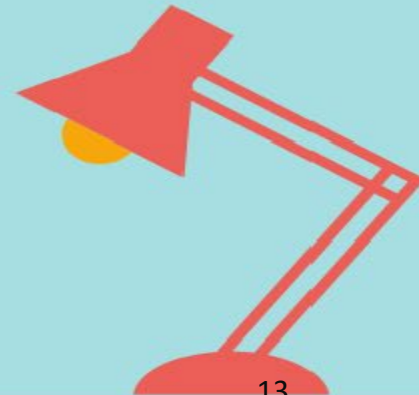
Command	Description
.show	Displays current settings
.databases	Provides database names and files
.quit	Quit sqlite3 program
.tables	Show current tables
.schema	Display schema of table
.header	Display the output table header
.mode	Select mode for the output table
.dump	Dump database in SQL text format



SQL(Structured Query Language) commands

- DDL-Data Definition Language
 - CREATE TABLE
 - DROP TABLE
 - ALTER TABLE
- DML-Data Manipulation Language
 - INSERT
 - UPDATE
 - DELETE
- DQL-Data Query Language
 - SELECT

case insensitive



SQL Examples

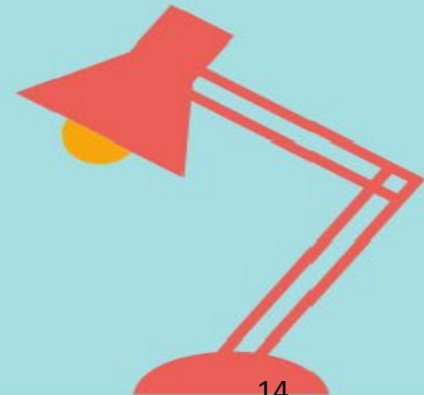
```
CREATE TABLE news (  
    news_id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    news_caption TEXT NOT NULL,  
    news_time DATETIME DEFAULT CURRENT_TIMESTAMP,  
    news_url TEXT NULL);
```

```
ALTER TABLE news ADD news_txt TEXT NULL;
```

```
INSERT INTO news (news_caption, news_url)  
VALUES ('N1', 'https://aaa.com/xxx111.aspx');
```

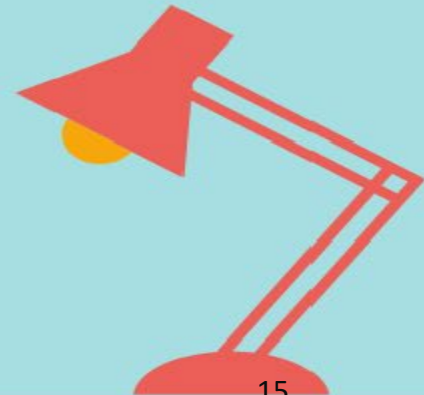
```
INSERT INTO news (news_caption, news_url)  
VALUES ('N1', 'https://aaa.com/xxx111.aspx');
```

```
SELECT * FROM news;
```



A Minimalist End-to-End Scrapy Tutorial

- This repo contains the code for my tutorial: A Minimalist End-to-End Scrapy Tutorial (<https://medium.com/p/11e350bcdec0>).
- The website to crawl is <http://quotes.toscrape.com>.



toscrape example

Quotes to Scrape

[Login](#)

"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."

by [Albert Einstein](#) (about)

Tags: [change](#) [deep-thoughts](#) [thinking](#) [world](#)

"It is our choices, Harry, that show what we truly are, far more than our abilities."

by [J.K. Rowling](#) (about)

Tags: [abilities](#) [choices](#)

"There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle."


by [Albert Einstein](#) (about)

Tags: [inspirational](#) [life](#) [live](#) [miracle](#) [miracles](#)



Top Ten tags

[love](#)[inspirational](#)[life](#)[humor](#)[books](#)[reading](#)[friendship](#)[friends](#)[truth](#)[simile](#)

Collecting data with Scrapy

GeeksforGeeks

Topic-wise Practice C++ Java Python Competitive Programming Machine Learning Web Development SDE Sheet Puzzles



Collecting data with Scrapy

Last Updated : 08 Sep, 2021

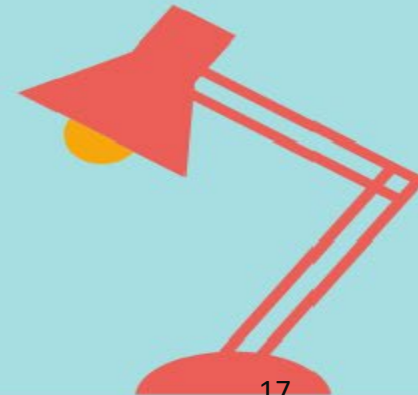
Prerequisites:

- [Scrapy](#)
- [SQLite3](#)

Scrapy is a web scraping library that is used to scrape, parse and collect web data. Now once our spider has scraped the data then it decides whether to:

- Keep the data.
- Drop the data or items.
- stop and store the processed data items.

Hence for all these functions, we are having a **pipelines.py** file which is used to handle scraped data through various components (known as a **class**) which are executed sequentially. In this article, we will be learning through the pipelines.py file, how it is used to **collect the data** scraped by scrapy using SQLite3 database language.





Thanks!

Q&A

