**撰寫網站爬行程序**

網站爬行程序的核心是遞迴。他必須從URL取得網頁內容，檢視網頁中的其他URL、取得該網頁；如此不停循環。

**遍歷單一網域**

維基的六度分離；目標是六次以內(包括原始主題)連結2個不相關的主題(例如:前者是維基網頁的相連，後者是同一部電影的演員)

舉例來說，Eric Idle與Brendan Fraser都演過Dudley Do-Right，後者與Kevin Bacon都演過The Air Breathe。此例中，Eric Idle與Kevin Bacon相關三個主題。從Eric Idle的維基網頁招出連結到Kevin Bacon的維基網頁最短的連結。

補充:

在維基百科裡面 屬性href 代表有連結

**爬整個網站(1-getWikiLinks.py)**

甚麼時候需要爬整個網站??

完整爬網站的一般作法是從頂層網頁(首頁)開始並搜尋該頁上的每個內部連結。

每個連結都爬過後在爬機中的連結。

**對遞迴的警告**

Python有預設的1000此遞回限制(程式遞迴呼叫次數)。由於維基的連結數量非 常大，程式可能會因為遞迴限制而停止，除非你能加上防止機制。

**跨整個網站蒐集資料(2-crawlWikipedia.py)**

網站爬行程序若只是從一頁跳到另一頁新的是無趣的。要讓他有用，必須對網頁做一些事情。

檢視如何建構蒐集標頭、第一段內容與編輯網頁的連結(如果有)的爬行程序。

* **所有標頭都在h1--> span 標籤下，他是網站唯一的的h1標籤。**
* **所有內容都在div#bodycontent標籤內。但若要第一段文字最好使用div#mw-content-text --> p(止選取第一段標籤)。**
* **編輯連結只出現在主題網頁。若有出現，他們會在li#ca-edit 🡪 span 🡪 a下的li#ca-edit當中**

**處裡重新導向**

重新導向(redirect)讓網頁伺服器指向另一個不同的網域名稱或URL的內容。重新導向分為2種

1.伺服器重新導向，URL在網頁載入前改變。(Python3.x的urllib函式庫會自動處裡)

2.用戶端重新導向

**跨網際網路爬行**

談到網站時，總是有人會問:”要怎麼做一個google?”。

“首先”要有很多錢買全世界最大的資料倉儲並存放到世界各地。

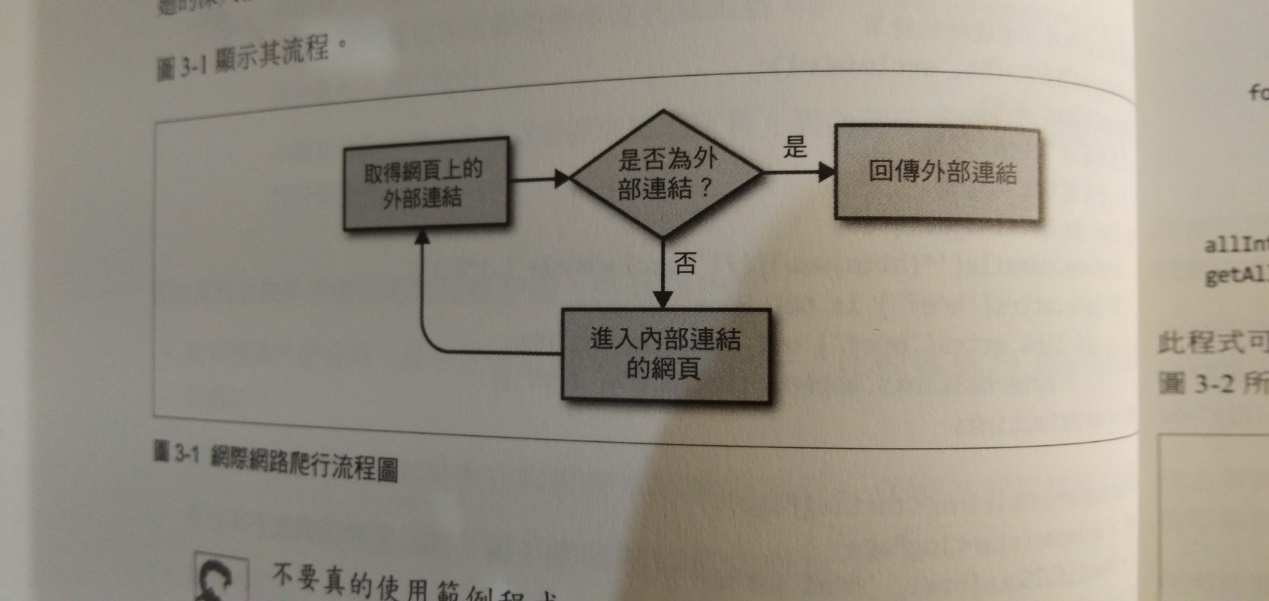
“其次”，你需要一個網站爬行程序。

**不要真的使用範例程式**

範例3(3-crawlSite.py)、4(4-getExternalLinks.py)、5(5-getAllExternalLinks.py)沒有寫出實務上的完整檢查與例外處理。舉例來說，若此爬行程序沒有遇到外部連結，則會一直跑到碰上Python的遞迴限制為止。

改善此程序的方法是與遞1張的例外處裡做結合。

V1.Chapter3 範例3、4流程圖



V1.Chapter3範例5流程圖