網路爬蟲與資料分析
靜態網頁解析

Instructor: 馬豪尚

HTML網頁基本架構

- 文件宣告DOCTYPE
- Html
  - 標示網頁的開始與結束，是一個網頁的根元素
- Head
  - 用來標示網頁標頭
  - 網頁編碼方式、標題、關鍵字、連結等
- Body
  - 網頁的主體

HTML基本範例

```
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
        <title>我的網頁</title>
</head>
<body>
        <h1>Hello, HTML5!</h1>
</body>
</html>
```

標籤(tag)與屬性(attribute)

- 標籤(tag)
  - 標示網頁上的內容或描述內容的性質
  - <head>、<body> 、<header> 、<p> 、 、<a> 、<table> 、<form> 、 <img>、 <video>等
- 屬性(attribute)
  - 超連結
    - Google首頁

屬性名稱　　屬性值　　內容

- 元素: 包含開始標籤、內容以及結束標籤

全域屬性 (global attributes)

- 所有的 HTML 元素都有的屬性，我們稱做全域屬性 (global attributes)，可以在所有的元素中使用
- **id元素唯一識別符號**:用來設定 HTML 元素的唯一識別符號 (identifier)，每個 HTML 元素的 id 需要是在整份 HTML 文件中獨一無二 (unique) 不可重複的，且一個元素只能有一個id。
  - 用作 <a> 連結的錨點名稱。例如點擊連結 <a href="#myid"> 會跳到 <tag id="myid"> 元素處
  - 用在 JavaScript 可以透過 id 存取該元素
  - 用在 CSS 可以用 id 當選擇器

Example

- <p id=“beauty">The most beautiful paragraph on this web. </p>

全域屬性 (global attributes)

- **class 元素類別名稱:**用來設定 HTML 元素的類別名稱 (class names)，每一個 HTML 元素可以有多個類別，你可以用空格分隔 (space-separated) 開不同的類別名稱。
  - 用在 JavaScript 可以透過 class 存取該元素
  - 用在 CSS 可以用 class 當選擇器 (selector)
- Example
  - Above point sounds a bit obvious. Remove/rewrite?

全域屬性 (global attributes)

- **style 樣式:** 用來直接設定該 HTML 元素的 CSS 樣式 (inline style)，而用 style 屬性設定的 CSS 優先權是最高的，會蓋過寫在 <style> 或外部樣式表中的樣式。
- Example
  - Hello World!

顯示結果：

Hello World!

HTML結構區塊標籤

標籤	用途	說明
<html>	HTML 文件的根元素	包含整個 HTML 文檔的內容。
<head>	文檔的頭部區域	包含了文檔的元數據，如 CSS 樣式鏈接、JavaScript 文件等。
<title>	定義文檔的標題	在瀏覽器的標題欄或頁籤上顯示。
<meta>	提供有關 HTML 文檔的元數據	用於指定頁面描述、關鍵詞、文檔作者、字符集等。
<link>	鏈接外部資源	用於鏈接外部資源，如 CSS 樣式表。
<script>	定義客戶端腳本（如 JavaScript）	用於在 HTML 文檔中嵌入或引用 JavaScript 代碼。
<body>	文檔的主體區域	包含網頁的所有可見內容，如文本、圖片、超連結等。
	區塊元素	用於創建一個邏輯容器，常用於 CSS 布局或 JavaScript 操作的目的。

HTML標籤

標籤	用途	說明
<h1> 到 <h6>	標題標籤	用於定義 HTML 標題，<h1> 是最大的標題，<h6> 是最小的標題。
<p>	段落標籤	用於定義文本的段落。
	行內元素	類似於 ，但用於行內元素的分組。
 	換行	在文本中插入一個換行，常用於段落或長文本內部。
<b>	粗體文本	使包含的文本顯示為粗體。
<i>	斜體文本	使包含的文本顯示為斜體。
	強調文本	使文本成為粗體，表示它的重要性更高。
	強調文本	使文本變為斜體，表示強調。
<a>	超連結標籤	用於定義超連結，可以鏈接到不同的頁面或頁面內的某個部分。
<blockquote>	引用區塊	用於定義長的引用，通常有縮進樣式。
<pre>	預格式化的文本區塊	用於顯示預格式化的文本，保留空格和換行。
<code>	程式碼	用於顯示一段程式碼，通常與 <pre> 標籤結合使用。
<img>	圖片標籤	用於在網頁中嵌入圖片。
<iframe>	內嵌框架	用於在當前文檔中嵌入另一個文檔，如嵌入 YouTube 影片、地圖等。

HTML標籤內常見屬性

href	<a>	指定超連結的目標 URL。	連結
src	<img>	指定圖像文件的路徑。	<img src=”image.jpg” alt=”描述”>
alt	<img>	為圖像提供替代文本。	<img src=”image.jpg” alt=”描述”>
name	<input>	為輸入元素指定名稱。	<input type=”text” name=”username”>
type	<input>	定義輸入元素的類型（如 text, radio）。	<input type=”checkbox”>
placeholder	<input>	提供輸入欄位的提示文字。	<input type=”text” placeholder=”姓名”>
value	<input>	定義輸入控件的初始值。	<input type=”submit” value=”提交”>

Requests向伺服器端GET請求

- 定義請求位置
  - url
- Requests用get方法來請求
  - request.get(url)
- 伺服器端回應屬性
  - text: 編碼的HTML標籤字串
  - contents: 沒有編碼的位元組資料，適用於非文字內容的請求
  - encoding: 取得HTML標籤字串的編碼
  - status_code: 伺服器回應狀態碼

Requests向伺服器端GET請求

- 帶有參數的請求
  - 直接將參數加在url網址之後
  - 使用params參數指定url參數值的字典
    - dic_params = {'name': 'Allen', 'grade': 100}
    - requests.get("url/get", params = dic_params)

Requests - User-agent and Cookie

- 有些網站的HTTP請求需要指定header參數或Cookie資料
- 輸入user-agent在header內
  - header= {'user-agent' : 'Allen'}
  - requests.get(url, headers=headers)
- 輸入Cookie資料
  - Cookies= dict('name'='Allen')
  - requests.get(url, cookies=cookies)

Requests向伺服器端POST請求

- 定義請求位置
  - url
- Requests用POST方法來請求，同時送出要傳送給伺服器的資料，例如表單欄位的輸入
  - post_data={'name':'Allen', 'grade': 100}
  - request.post(url, post_data)

靜態網頁分析

- 用requests取回HTML網頁內容
- 搭配BeautifulSoup套件
  - 解析及取得HTML原始碼各個標籤的元素資料
  - pip install bs4
- 載入套件模組
  - from bs4 import BeautifulSoup

BeautifulSoup解析器

› Beautiful Soup支持Python標準庫中的HTML解析器，還支持一些第三方的解析器

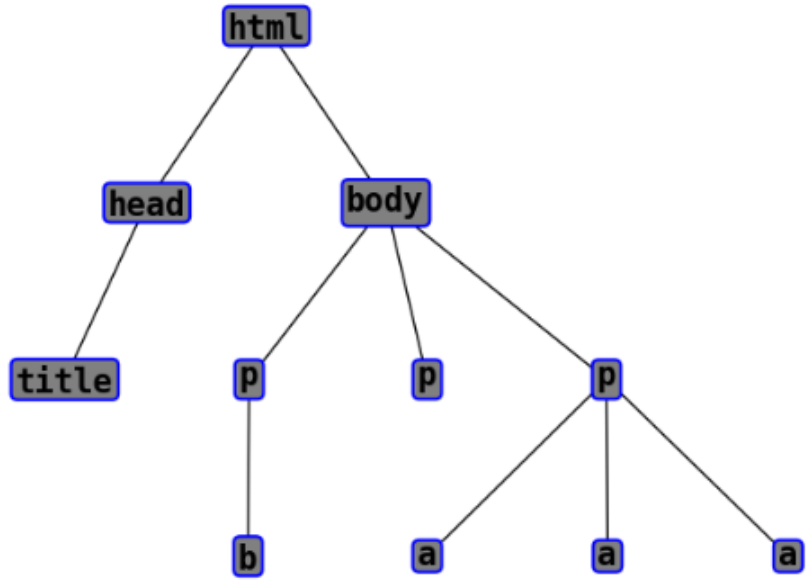
解析器	使用方法	優勢	劣勢
Python標準庫	BeautifulSoup(markup, "html.parser")	執行速度適中 文檔容錯能力強	Python 2.7.3 or 3.2.2前的版本中文檔容錯能力差
lxml HTML 解析器	BeautifulSoup(markup, "lxml")	速度快 文檔容錯能力強	需要安裝C語言庫
lxml XML 解析器	BeautifulSoup(markup, ["lxml-xml"]) BeautifulSoup(markup, "xml")	速度快 唯一支援XML的解析器	需要安裝C語言庫
html5lib解析器	BeautifulSoup(markup, "html5lib")	最好的容錯性 以瀏覽器的方式解析文檔 產生HTML5格式的文檔	速度慢 不依賴外部擴展

BeautifulSoup

- 安裝解析器
  - lxml HTML 解析器
    - pip install lxml
  - html5lib解析器
    - pip install html5lib

BeautifulSoup 物件的種類

- Beautiful Soup將HTML文檔轉換成一個樹形結構，每個節點都是一個物件
  - BeautifulSoup
    - soup = BeautifulSoup(content, "lxml")
    - 第一個參數為含有HTML標籤的文字內容
    - 第二個參數為解析器的名稱
  - Tag
    - 與XML或HTML原始文檔中的tag相同
  - NavigableString
    - 可以遍歷的字符串
  - Comment
    - 註釋及特殊字符串



BeautifulSoup 物件

- Beautiful Soup提供了許多操作和遍歷tag子節點
- 存取某個tag的方法
  - soup.tag名稱
    - tag = soup.a → 獲取解析文檔內的第一個a標籤
    - tag = soup.body.p → 獲取解析文檔內的body標籤底下的第一個p標籤
  - 使用find()
    - soup.find('tag名稱')
    - 只能存取第一個名為'tag名稱'的節點
- 要存取/查詢所有指定名稱的子節點(tag)
  - soup("tag名稱")
    - 返回一個列表為符合tag名稱的所有tag和內容

BeautifulSoup 物件

- find_all(name, attrs, string, recursive, **kwargs)
  - name 參數可以查找所有名字為 name 的tag
    - soup.find_all("title")
  - attrs 參數搜尋時可以把該參數當作指定名字tag的”屬性值”來搜尋
    - 可以使用的屬性值支援字符串、正規表達式、列表、True
    - soup.find_all(id='link2')
    - soup.find_all(href=re.compile("elsie"), id='link1')
    - 有些tag屬性在搜索不能使用,比如HTML5中的 data-* 屬性
  - string 參數可以搜尋文檔中的符合字符串的內容
    - soup.find_all(string="Elsie")
    - string 參數能夠支援字符串、正規表達式、列表、True

BeautifulSoup 物件

- find_all(name, attrs, string, recursive, **kwargs)
  - recursive=True/False
    - 只想搜索tag的直接子節點，使用參數 recursive=False
    - 預設搜尋tag的所有子孫節點，recursive=True
  - limit=數字
    - 搜尋到的結果數量達到 limit 的限制時，就停止搜尋返回結果
- find_all可以使用css的類別方式來查詢
  - soup.find_all("a", class_="sister")
  - 支援字符串、正規表達式、方法或 True

BeautifulSoup 物件

- 使用參數CSS選擇器的搜尋方法
  - select(“選擇器”)
    - soup.select(“選擇器”) → 搜尋所有符合該選擇器的節點(tag)
    - soup.select(“#id”) → 若使用id為選擇器則選擇器名稱為#id
    - tag.select(“選擇器”) → 搜尋tag內符合該選擇器的子節點
    - 以上都會返回一個列表，包含所有符合的節點內容
  - select_one(“選擇器”)
    - soup.select_one(“選擇器”) → 搜尋第一個符合該選擇器的節點(tag)
    - soup.select_one(“#id”) → 若使用id為選擇器則選擇器名稱為#id
    - tag.select_one(“選擇器”) → 搜尋tag內符合該選擇器的第一個子節點

BeautifulSoup 物件搜尋方法小結

搜尋方法	說明
select_one()	使用參數CSS選擇器字串搜尋HTML標籤，返回第一個符合的HTML標籤物件
select()	使用參數CSS選擇器字串搜尋HTML標籤，返回所有符合的HTML標籤物件的串列
find()	使用參數的標籤名稱或屬性值來搜尋HTML標籤，返回第一個符合的標籤物件
find_all()	使用參數的標籤名稱或屬性值來搜尋HTML標籤，返回所有符合的HTML標籤物件的串列
soup.tag名稱	返回一個符合該名稱的標籤物件
soup(“tag名稱”)	返回一個串列包含所有符合該名稱的標籤物件

練習

- 用requests請求fChart 程式設計教學工具網頁
  - https://fchart.github.io/
- 用BeautifulSoup套件解析網站內容
  - 解析出所有圖片的標籤並存成csv(以流水序號為index)
  - 解析出所有含有"編輯器"的文章段落並將這些段落以長度排序儲存成csv(以流水序號為index)