# 網路爬蟲與資料分析
## 靜態網頁解析

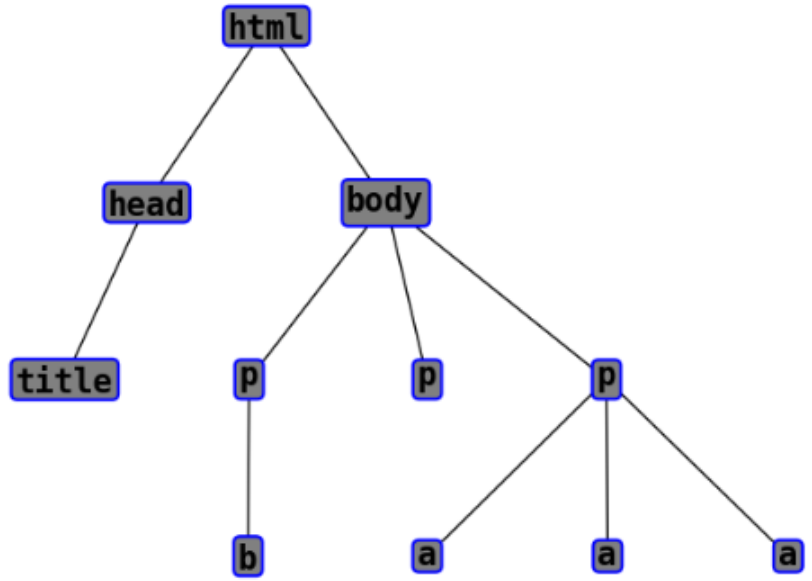
Instructor: 馬豪尚

# 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 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 Tag 物件屬性存取

- Tag與XML或HTML原始文檔中的tag相同，
- tag.name
  - 每個tag都有自己的名稱，透過這個name屬性來獲取
  - 可以改變tag的名稱但會影響整個透過當前Beautiful Soup物件解析的HTML文檔樹
    - tag.name = "新的名稱"
- tag.text
  - 取得當前tag的內容

# BeautifulSoup Tag 物件屬性存取

- tag.attrs : 一個tag可能有很多個屬性，tag的屬性操作方法與字典相同
  - 一個屬性的情況
    - <b class="boldest"> 有一個 "class" 的屬性，其值為 "boldest"
      - {'class': ' boldest'}
  - 多個屬性的情況
    - 有一個 "class" 的屬性，其值為 "box"，和另一個"id"屬性，值為"one"
    - tag.attrs → 返回所有屬性名稱和值的字典
      - {'class': ' box', 'id': 'one'}

# BeautifulSoup Tag 物件屬性存取

- HTML多值屬性，一個屬性內含有多個值(用空白隔開)
  - 最常見的多值的屬性是 class
  - 有一個 "class" 的屬性，其值為 "one" 和"box"兩個
  - 操作和單一屬性值的時候一樣，但是會用列表返回
    - tag[屬性名稱] → ex. tag['class']
    - 返回 ["one ", " box"]的列表
  - 如果該屬性為不支援包含多值的屬性，返回會和單一屬性值一樣為字串

# BeautifulSoup Tag 物件屬性存取函式

- tag.get("屬性名稱", None)
  - 第一個參數為指定屬性，意思是取得當前tag內指定屬性的值
  - 第二個參數為，若沒有符合的屬性而返回的值
- tag.getText()
  - 取得<a>標籤的連結文字

# BeautifulSoup Tag 物件操作

- 文檔樹中tag的屬性可以被增加，刪除或修改，操作方式採用python字典的操作方式
- 存取屬性
  - tag[‘屬性名稱’]
- 增加或修改屬性
  - tag[‘屬性名稱’] = 屬性值 → ex. tag['id'] = 1
- 刪除屬性
  - del tag[‘屬性名稱’] → ex. del tag['class']

# BeautifulSoup Tag 物件子節點

- tag.contents
  - tag的 .contents 屬性可以將tag的子節點以列表的方式返回
- tag.children
  - tag的 .children 生成器，可以對tag的子節點進行循環
  - for child in tag.children:
- tag.descendants
  - .contents 和 .children 屬性僅包含tag的直接子節點
  - .descendants 屬性會返回一個可迭代的物件
  - 得到該節點的所有子孫節點

# BeautifulSoup Tag 物件子節點

- tag.string
  - tag僅有一個子節點可以用tag.string來存取
  - tag只有一個 NavigableString 類型子節點，也可以用tag.string來存取
  - tag包含了多個子節點，tag就無法確定 .string 方法應該存取哪個子節點的內容，就會返回None

# BeautifulSoup Tag 物件子節點

- tag.strings
  - tag中包含多個字符串，可以使用 .strings 來循環獲取
    - for string in soup.strings:
- tag.stripped_strings
  - 輸出的字符串中可能包含了很多空格或空行，使用.stripped_strings這個方法來去除
    - for string in soup.stripped_strings:

# BeautifulSoup Tag 物件父節點

- tag.parent
  - 獲取某個tag的父節點
  - 最頂層節點的父節點就是BeautifulSoup物件
  - BeautifulSoup 物件的父節點會返回None
- tag.parents
  - .parents 屬性可以返回一個可迭代的物件
  - 得到該節點的所有父輩節點

# BeautifulSoup Tag 物件兄弟節點

- tag.next_sibling
  - 查詢指定一個tag的下一個兄弟節點
    - tag = soup.b
    - tag.next_sibling
- tag.previous_sibling
  - 查詢指定一個tag的上一個兄弟節點
    - tag = soup.b
    - tag.previous_sibling
- tag.next_siblings 和 tag.previous_siblings
  - 透過 .next_siblings 和 .previous_siblings 屬性可以對當前節點的兄弟節點返回一個可迭代的物件

# BeautifulSoup NavigableString

- NavigableString
  - 被定義為可以迭代遍歷的字符串，字符串常被包含在tag內
  - NavigableString沒有子節點的方法可以使用，因為沒有子節點
  - tag.string → 存取這個標籤內的字符串

# BeautifulSoup修改文檔樹

- 修改tag的名稱和屬性
  - tag.name = “新名稱"
  - tag[‘屬性’] = ‘新屬性值’
- 刪除屬性
  - del tag[‘屬性']
- 修改.string
  - tag.string = “新的String值”
- 創建一個新tag
  - soup.new_tag(“tag name”, 屬性=“屬性值”)
- 創建一段新文字內容
  - soup.new_string(“文字內容")

# BeautifulSoup修改文檔樹

- 增加tag內的文字內容
  - tag.append(“增加的內容”)
- 插入tag或文字內容
  - insert() : 插入到指定位置
    - tag.insert(指定位置, 文字內容)
  - insert_before() : 插入到當前tag之前
    - 當前tag.insert_before(soup.new_tag(“tag name”, 屬性=“屬性值”))
  - insert_after() : 插入到當前tag之後
    - 當前tag.insert_after(soup.new_string(文字內容))

# 練習

- 用requests請求ETtoday新聞網站內容
  - https://www.ettoday.net/
- 用BeautifulSoup套件解析網站內容
  - 將即時新聞清單解析出來包含以下欄位的內容
    - 新聞類別、 新聞標題、url網址、時間(date)
  - 用新聞類別排序所有新聞(同一類別照新聞時間排序)，並存到csv成為一個表格