# 抓取網頁的最佳語言：Python | 程式設計 遇上 小提琴

星期六, 1月 7 2012, 4:40 上午




← 測試的好幫手: 虛擬機器

WTF : 好萊塢版七龍珠 →


## 抓取網頁的最佳語言：Python

Posted on 2008 年 10 月 05 日 by victor

### 最初

最早我用C/C++語言慢慢寫抓網頁的用它來抓網頁真的是程式，一開始甚至打算自己寫抓取網頁的函式庫，想說當做練習，可是HTTP協定 雖然不難，可是煩，要處理的細節太多了，後來受不了，轉而使用現成的Library：cUrl，但是C/C++語言開發這類東西的效率實在太慢了，我的程式不停的修改、不停的修改，光是編譯的時間就吃掉了不知道多少，字串的處理C/C++ 沒有內建正規表示法或一些好用的字串函數之類的，處理起來也礙手礙腳，當時，我想將我寫好的函數庫寫成能讓Lua呼叫的形式，或著甚是C/C++來呼叫Lua，因為C/C++有很多細節要處理，Memory leak有的沒有的雜事，我想要的只是專注在寫抓取網頁的程式，因此用Lua包裝似乎是不錯的選擇，但是開發時間太久了，事情一直沒有變好

### 直到

我下了一個結論，C/C++不適合寫抓取網頁的程式，我開始思考我需要什麼，我想我既然要包裝成其它語言將細節藏起來，為何不直接使用script語言? 我最早一直擔心的是效率的問題，但是到後來想想反正真正沒效率的部份包給C/C++去做事實上沒有太大的差別，而且又有動態語言的彈性、除錯上的方便等等好處，何樂不為? 於是我開始尋找一款合適的語言

### Perl 如何?

很早以前我有用Perl寫過一些CGI程式、留言版、網站管理系統、文章管理系統等等，有人說Perl是只能寫一次的語言，它有很多很簡短的符號所構成的表示法，可讀性不是很好，模組化設計也沒有非常好的支援，OO也是一樣，新版的Perl遲遲沒有推出，似乎已經有點變成遺產的感覺，或許是上面的理由還是偏見，總而言之我不喜歡Perl

### PHP?

做為一個以網頁為主要用途的語言，拿來當做其它用途總有種不太合適的感覺，從它的語法來看，很明顯是參考C語言、Perl等等而來的，但是卻沒有加以改進，我個人認為它可能沒有預料到PHP居然會紅成這樣，變成網頁程式設計的主流語言，後來有很多缺點就變得顯而易見，不夠嚴僅的語法、不夠好的模組化設計、不良的OO支援、容


**關於作者**

我是Victor Lin，Now.in的創辦人，興趣是程式設計，Python目前是我最喜歡的語言，從國一開始寫程式到現在已經有十個年頭，不過還有很多要學習，除此之外偶爾用小提琴製造一些噪音也是我的興趣之一 E-Mail:
bornstub@Gmail.com





**標籤**

Android C++ C/C++ CMS Django Game Game design HGE i18n

易寫出安全性有問題的程式等等，命名空間也是它一大缺點之一，光是看到一大堆前綴字開頭的函數就有種倒胃口的感覺，有人說

> *PHP is the BASIC of the 21st century*

在這個影片裡，總合種種理由，做為抓取網頁的用途，PHP出局

## Lua

Lua做為輕量級的語言相當的優秀，可是你不會想用Lua來寫大型的程式，我也不會想這麼做，它語言的設計都是以速度為優先考量，寫起來並不怎麼順手的感覺，再加上目前的資源不多，可能很多東西都得自行包裝，這樣就和我原先想做的事是一樣的，因此不考慮Lua

## Java

Java是和網路一起成長的程式語言，做為抓取網頁的用途，它絕對有能力勝任，但是…，我嫌它太囉唆了，還有太癡肥，當一款語言太囉唆和太癡肥往往會令人討厭，歐! 想到Java我就想起eclipse在我那台只有256扣掉分給顯示記憶體的筆電上執行的情況，讓我想把電腦砸掉，不好意思，我不喜歡Java

在前面的影片裡的老兄一樣也有提到，有興趣可以看看

> *Java is the COBOL of the 21st century*

## Python

最後，我在PTT的程式設計討論版上描述了我的需求，有人推文說 Python，我抓了抓頭髮，Python? WTF? 這是什麼? 我從來沒有聽過這款語言，於是上網找了一下資料，和問了一些問題，發現這款語言正是我想要的，它很容易被擴充，因此效能不足可以用C/C++補強，你想得到的函式庫幾乎都已經有人寫好了，光從下載網頁這件工作來看，它的標準函式庫已經有了這樣的功能，你覺得不夠好還有其它很多的選擇，開箱即用的哲學，讓安裝函式庫非常簡單，不像C/C++的編譯惡夢讓你抓光頭髮，而它最優秀的地方之一就是它的可讀性，寫起來相當順手、優雅，讀起來也一樣順眼，重要的是很有趣，那麼開發大型的程式呢? script語言常見的問題就是對於開發大型程式來說很不適合，但是Python卻不是如此，良好的OO、模組化等等它都有良好的支援，再加上Google也是Python的愛用者，YouTube也是用Python開發的，有了這些大咖背書，證明這款語言的確是相當優秀，在決定使用Python之後我就立刻訂購了一本Learning Python，開始學習Python

## 愛上Python

Python並沒有讓我失望，能用Python寫的東西都不太想用C/C++去寫，開發效率非常高、寫起來很順手、豐富的資源，讓我覺得這真的是優秀的語言，它的確很適合拿來抓取網頁，不過抓取網頁還有更多東西要考慮

## Twisted

用Python抓取網頁的HTML只是小菜一盤，用Python標準函數庫就辦得到，但不是那麼好用，最後我發現了Twisted，就改用Twisted來抓網頁，它有優秀的非同步事件驅動的架構，常見的協定都已經有實做，包括HTTP、SMTP等等，用它來抓網頁真的是再容易不過了

```
getPage("http://www.google.com").addCallback(printPage)
```


Linux Logging lxml MySQL note now.in Open source PHP Plone Python Server tool TurboGears twisted undefined behavior WebFaction WTF 主機商 分享 嘴砲 奇怪 心得 抓取網頁 測試 破爛英文 程式設計 網站 腦殘 虛擬機器 設計 資安 軟體工程 遊戲 遊戲設計 開源 音樂

**彙整**

- 2011 年 十二月
- 2011 年 十一月
- 2011 年 十月
- 2011 年 九月
- 2011 年 八月
- 2011 年 七月
- 2011 年 六月
- 2011 年 五月
- 2011 年 三月
- 2011 年 二月
- 2011 年 一月
- 2010 年 十月
- 2010 年 八月
- 2010 年 七月
- 2010 年 五月
- 2010 年 四月
- 2010 年 三月
- 2010 年 一月
- 2009 年 十二月
- 2009 年 十一月
- 2009 年 十月
- 2009 年 九月
- 2009 年 八月
- 2009 年 七月
- 2009 年 六月
- 2009 年 三月
- 2009 年 二月
- 2009 年 一月
- 2008 年 十二月
- 2008 年 十一月
- 2008 年 十月
- 2008 年 九月

**分類**

- Android
- 小提琴
- 專題
- 中文文章
- 作品
- 分享
- 嘴砲
- 哇咧咧
- 問題
- C/C++
- 筆記
- 網站
- 病毒
- 無用
- English Articles
- 音樂
- 遊戲
- 遊戲設計
- 設計
- 資訊安全
- 英文
- Javascripts
- Linux
- Python
- Uncategorized
- Unix-Like
- WTF

是的，一行就可以抓網頁，夠簡單吧，而且你想要傳POST或GET等參數，或是修改HTTP的header都沒有問題

## BeautifulSoup

抓網頁事實上不是什麼難事，解析HTML要來得更麻煩，最初使用Python的標準函式庫內建的HTMLParser來解析網頁，但是功能太陽春，加上最頭痛的問題是，大部份的網頁都沒有完全尊照標準來寫，各種莫明奇妙的錯誤令人想要找出那個寫網頁的人痛打他一頓，為了解決容錯的問題，一開始我使用BeautifulSoup來抓取網頁，它是以容錯著名的HTML Parser，但是，它的效率很差，又或著說，找到目標HTML標籤的方式很沒效率，一般都用find等方式來找到所要的標籤

```
soup.find('div', dict(id='content'))
```

它真的很沒效率，當你抓取大一點的網頁時，多塞幾個一起抓和解析，你就會看見你的CPU使用率永遠是滿的狀態，原本我預計抓網頁的瓶頸都會落在網路IO上面，但是用它來抓取網頁卻超出我預料，沒想到它會這麼吃重，於是沒辦法，我開始尋找更好的選擇

## lxml

我找到一個Blog的文章：Python HTML Parser Performance，介紹了Python各種Parser的效能，效能最亮眼的，就是lxml，我最初擔心的是找到資料標籤會不會很困難，但是我發現它支援xpath，就試著改寫原本BeautifulSoup用find等等函數寫的尋找標籤程式，發現xpath遠比那種方式來得好用太多了，而且效率好太多了，BeautifulSoup的find極度的沒有效率，大部份的CPU時間都耗在一堆find函數走訪HTML樹上，而xpath篩選標籤的方式來得有效率多了，以下舉幾個我實際用在抓取網頁的案子中的例子

```
def getNextPageLink(self, tree):
    """Get next page link

    @param tree: tree to get link
    @return: Return url of next page, if there is no next page, return N
    """
    paging = tree.xpath("//span[@class='paging']")
    if paging:
        links = paging[0].xpath("./a[(text(), '%s')]" % self.localText['
        if links:
            return str(links[0].get('href'))
    return None
```

```
listPrice = tree.xpath("//*[@class='priceBlockLabel']/following-sibling:
if listPrice:
    detail['listPrice'] = self.stripMoney(listPrice[0].text)
```


- 測試
- 數學筆記

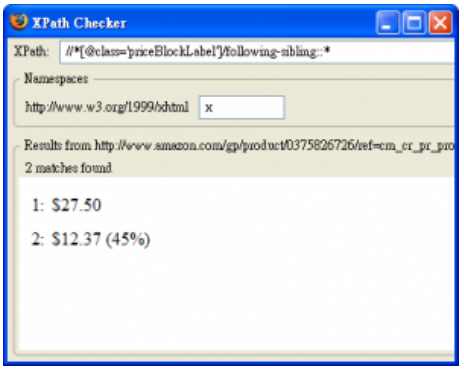
**近期迴響**

- tofu 在 李大師您多久沒寫程式了？一百個你不應該繼續用Dev C++ 4.9.9.2 的理由
- victor 在 新世紀通訊函式庫 – ZeroMQ
- victor 在 新世紀通訊函式庫 – ZeroMQ
- thomasy 在 新世紀通訊函式庫 – ZeroMQ
- tamalu 在 李大師教初學者用C語言? MIT使用Python

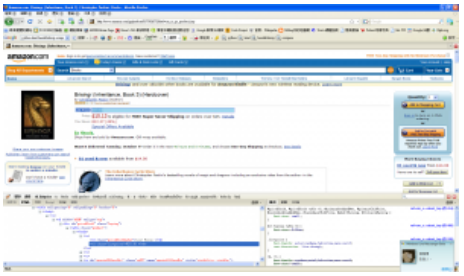
原本使用BeautifulSoup在尋找標籤遇到麻煩的走訪羅輯上的問題還得寫程式解決，xpath本身就有豐富的語法可以提供各種篩選的條件，羅輯從程式碼被移到了xpath語法上，有了這樣的語法，尋找目標標籤輕鬆了許多，而且效率也很好，從此我就和BeautifulSoup說再見，改用lxml來找標籤

## 配合FireFox的工具

如果有一些工具可以幫助寫解析網頁的程式該有多好，這也是我希望能有的，使用了xpath之後，我找到了FireFox的插件，XPath checker等xpath的工具，可以先用它來確定抓到的元素是正確的，然後FireBug在檢視網頁結構上也有很大的幫助




FireFox插件XPath checker畫面



使用FireBug檢視網頁元素

## 結論

就目前一路走過來的經驗來看，抓取網頁Python的確是最佳的選擇，不過我們到目前為止我們都只討論到工具，事實上還有設計上的問題要解決，留在下一次寫

**書籤:**

This entry was posted in 中文文章, Python and tagged 心得, 程式設計, Python, 抓取網頁. Bookmark the permalink.

← 測試的好幫手: 虛擬機器

WTF : 好萊塢版七龍珠 →

### 33 Responses to *抓取網頁的最佳語言：Python*

日落 *says:*

2008 年 10 月 05 日 at 5:35 下午

Java 拿來抓網頁資料絕對很方便
因為他的處理字串的函式很多
但是就是……肥

victor *says:*
2008 年 10 月 05 日 at 5:38 下午

你忘了說 還有很囉唆吧= ="
相較之下Python就簡單可愛

科科

錯字 *says:*
2008 年 10 月 05 日 at 8:55 下午

用它來抓網頁真的是"在"容易不過了,
另…雖然我沒有強到可以置評其他人的地步，但我覺得…沒有不好的語言，只有用不對地方而已，以上。

victor *says:*
2008 年 10 月 05 日 at 9:15 下午

感謝，已修正，我也這樣覺得，每種語言都有其特性、哲學，對於不同用途也有不同的強弱，而每個人也有喜好不同，所以語言本身沒有好壞，端看如何使用它 :P

schneider *says:*
2008 年 10 月 07 日 at 11:24 下午

最近打算開始學python，也是以抓網頁資料下來處理作為入門，國內python的資料不多，你的文章讓我受益不少，感謝 :D

Brian *says:*
2008 年 10 月 08 日 at 1:34 上午

文中的 memory lack 可能是指 memory leak。

victor *says:*
2008 年 10 月 08 日 at 10:04 上午

已修正 我好像很容易把leak打成lack XD

yen3 *says:*
2008 年 10 月 08 日 at 10:50 上午

C++ Boost Library建議可以看看，雖然在TR2才會有對http之類的支援，但是在boost裡已有regular expression了。
當然，我不否認，Python是一個極具威力的語言，但是通常在與很多語言的比較上，我通常會語帶保留，因為我不是該語言的專家，我沒有辦法下這麼多定論。第一印象不代表全部。

冒犯了。

**victor** *says:*

2008 年 10 月 08 日 at 12:57 下午

Boost有Regular沒有錯，但整體上看來還是不適合，有很多簡單的工作全部都寫regular expression會很辛苦，相較之下python有切片的語法，在處理字串就特別的方便

Boost應該沒有支援Http吧? 你說的是asio吧? 它應該只是非同步的IO網路函式庫，並沒有實做http，如果要http的話有一款叫pion的library是架構在asio上的，可以參考看看

事實上，我說的最佳有前提的，是就我目前所認識到的這些東西，還有個人喜好，當然沒什麼是絕對的最佳，對我來說，目前就這樣是最佳的 :D

Joe *says:*

2008 年 12 月 08 日 at 11:21 上午

沒有Regular在網頁處理上簡直是廢物，
切片?你切到最後可讀性就沒了，天知道你切的第幾片是你要的。
網頁處理可不單單只是把網頁抓下來而已
僅僅是這樣子的話
拿HTTP 協定直接handshake不是更行？
去蕪存菁才是重點

**victor** *says:*

2008 年 12 月 08 日 at 11:47 上午

問題是Python也有Regular阿= ="
C++也有Regular，所以重點不在Regular
而反而是其它一些簡單的工作之類的
現在哪款語言沒有regular expression可以用?
我不懂你指的廢物是指什麼? 都有Regular expression可以用阿 囧

為什麼殺雞要用牛刀? 如果只是把文字拆成兩三個部份，用regular根本是增加問題而已，當然，如果複雜到一定程度，用切片反而是更麻煩，所以在適當的情況用適合的工具才是正確的選擇，除此之外Python有C++可能要自己動手寫的好用函數，例如strip去頭尾空白，startswith和endswith比較字串開頭字串結尾，還有諸如此類很多麻煩的地方

為什麼不用HTTP直接抓取網頁?因為有太多細節要處理，光是HTTP協定的實作就可以弄到吐血，舉個例子，光是重導就有很多種，你跟還是不跟? 光是跟又有很多麻煩的問題，遇到無限遞迴的重導，沒處理好的話你的程式就永遠在那邊重導，因此光是把HTTP協定弄好頭髮已經白一半了，目的是要抓網頁，不是要重造輪子

你說的沒錯，去蕪存菁的確才是重點，所以我才說我不想理那些細節，我只想專心在抓取網頁上面，還有你可能沒仔細讀，在用Regular或切片解析之前，還有一個lxml或BeautifulSoup，用於把網頁parse成tree，然後再用xpath去找標籤，例如我要找網頁的title可能會像這樣子

```
title = tree.xpath('//title/text()')[0]
```

title的內容就直接是 裡面的xxxx，我想你可能誤會了，以為我是用切片去切整

個網頁的html? 怎麼可能= ="，我猜你指的沒有regular expression是廢物，是因為你用regular expression去解析網頁，但是那很沒效率，而且又很麻煩，可讀性也差到不行，維護上面對整堆的regular expression更是惡夢

以解析成語法樹之後來說，到了這個步驟，幾乎都只剩下簡單的文字要處理，例如

作者: John

像這種你要用Regular expression嗎? 有這麼嚴重嗎= =? 除非更複雜一些的我才考慮要使用re

所以…不好意思，我想你的質疑是來自於誤解，可以請你仔細再讀一次嗎? 還是我寫得不清楚，有需要解釋的地方嗎?

---

Joe *says:*

2008 年 12 月 08 日 at 12:02 下午

我沒指定特定的語言來說
我是泛指沒有Regular能力的語言根本就不要堤網頁處理這件事了

另外針對切片法來評論
是對應到前面說的可讀性
切片法的可讀性真的是OOXXOOXX
(當然你假如拿小網頁當範例我也沒話說)

此外，
Regular寫的好，讀的到的資料都會很準確
讀不到也很乾脆的跟你說讀不到
而不像切片法可能有這個也可以、那個也可以的現象

---

victor *says:*

2008 年 12 月 08 日 at 12:35 下午

我提到的切片只是某些地方會用到，並不是表示我全部的解析都靠切片達成，答案是，切片也不常用到，因為tree解析出來資料幾乎都出來了，會需要切片的只有內容文字也有格式的情況，那樣的情況實在不多

你指的小網頁範例是多小? 抓取amazon.com書店的書籍資料夠大了嗎?
我給你看實際的例子 amazon.com的書標題後面
Twilight (The Twilight Saga, Book 1) (Paperback)
會有一個()括起來的書本裝訂方式
在這裡我這樣寫

```
def getTitle(tree, localText=usaText):
"""Get title and binding of book

@param tree: Html tree to get title
@param localText: Local text table
@return: title, binding
"""
title = tree.xpath("//*[@id='btAsinTitle']")
if title:
title = unicode(title[0].text.strip())
```

```
bookTitle = title[:title.rfind('(')].strip()
binding = title[title.rfind('(')+1:title.rfind(')')].strip()
return bookTitle, binding
return None, None
```

你有讀不懂嗎? 夠清楚了吧? 倒著找(和)之間內的切片

另一個片段我用到regular expression長這樣，因為用切片會更麻煩

```
pattern = re.compile('([^;]+)(;(.+))?\\((.+?)\\)')
publisherField = unicode(detailMap.get(localText['publisher']))
publisher, _, edition, publishedDate =
pattern.match(publisherField).groups()

publisher = publisher.strip()
edition = edition.strip() if edition is not None else None
publishedDate = publishedDate.strip()
```

看那個regular expression，請問你可以在30秒之內告訴我它的目的嗎?

答案是這樣:

Course Technology; 7 edition (February 26, 2007)

amazon.com的出版社欄位包含3個部份

出版社；版本 (出版日期)

版本有可能會省略不見，所以我不寫出來，兩個可讀性請問哪個比較好= =?
而且這些只是少部份需要處理文字的，大部份抓到標籤文字都沒有另外的格式需要特別處理

例如

```
def getPrice(tree, localText=usaText):
"""Get list pirce and price of book

@param tree: Html tree to get price
@param localText: Local text table
@return: list price, price
"""
def strip(text):
return text.replace(',', '').replace(localText['dollorSign'], '').strip()
# get list price
listPrice = tree.xpath("//*[@class='priceBlockLabel']/following-sibling::*")
if listPrice:
listPrice = strip(listPrice[0].text)
listPrice = float(listPrice)
else:
listPrice = None

# get price
priceTd = tree.xpath("//td[@class='priceBlockLabelPrice']/following-
sibling::*")
if priceTd:
price = priceTd[0].xpath(".//b[@class='priceLarge']")
price = strip(price[0].text)
price = float(price)
```

else:
price = None
return listPrice, price

價格在parse階段早就被標籤分好了，根本不需要另外去做什麼處理，八成的資料都不用額外解析，而如你所見，如果用regular expression去寫，很常會出現像火星文一樣的語法，有很多字元需要跳脫之類的，一堆鬼符號揪結在一起，那個可讀性哪裡好= =|| 我真的完全看不出來

再者，你的工作大部份都在想要怎樣的Regular才能抓到正確的資料，有些部份很麻煩，你要先找到特別的標的語法，然後再靠這個語法去找到相對的html片段的關係，很多模陵兩可的情況會讓你嘔出血來，因為你的regular沒辦法分出上下文之間的關係

但是解析成文法樹的話，所需要思考的層面從語法被提升到語法樹之間的關係，可以更輕易地專注在你要找的資料上，只要能抓到標籤，就等於找到了資料

如果還是覺得全用regular expression從html解析開始做比較好的話，我們可以來比較看看，我已經有用python寫好的amazon抓取書籍資料的程式，你也寫一個全用regular expression的版本來比比看，可讀性、效能、維護上等等各層面上的優缺點，或許是我太嫩，我真的不懂得要怎樣寫出看起來不像火星文又好維護的regular expression用來抓取網頁

---

Joe *says:*
2008 年 12 月 08 日 at 1:08 下午

我想有幾點針對標題和切片法、還有regular expression

1.這篇文章很明顯的是比較各個程式語言
所以當你試著用切片法寫在其他程式語言的時候，有沒有那麼好用？
2.Regular在你看來是火星文，但是其他人可能不是。我想這是能力問題，我認為沒必要特別以自己的短處來當做理由。
3.Regular expression可以抓的很簡單、很直接，有些動作可能一次一個網頁內容就可以抓完。

---

Joe *says:*
2008 年 12 月 08 日 at 1:13 下午

簡單的一個例子
請你把tw.yahoo.com首頁所有的a link找出來
這類越亂的程式就可以看出切片法的困難了
因為你找不出順序規律

amazon.com的資料這麼整齊
我想考驗不出能力來吧
數著下一行或是下一字也可以把資料找出來

---

victor *says:*
2008 年 12 月 08 日 at 1:26 下午

你根本還沒搞清處狀況嗎? 我說切片只有少數情況會用到，你說的抓tw.yahoo.com首頁的所有link就這樣而以… 你還想說什麼?

我笑了，amazon.com太整齊考驗不出能力來，不然要什麼才考驗得出能力?
你要不要用regular expression寫出來抓amazon.com資料來給我看再來說嘴?
出一張嘴誰都會
##############
import cStringIO

import pycurl
from lxml import etree

import http_util

c, body = http_util.curl('http://tw.yahoo.com')
c.perform()

html = body.getvalue()
parser = etree.HTMLParser(encoding='utf8')
tree = etree.parse(cStringIO.StringIO(html), parser)

links = tree.xpath('//a')
for link in links:
print unicode(link.get('href'))

---

victor *says:*
2008 年 12 月 08 日 at 1:38 下午

可讀性本來就帶有主觀色彩，你或許覺得regular expression你很強，又如何? 可讀性又是什麼? 是這樣符號揪在一起，用能力來決定讀不讀得懂叫做可讀性好? 我笑了，這那門子的可讀性??? 符號揪在一起，要花很多時間讀，能力強的讀得懂? 這不是可讀性差是什麼? 那還只是算簡單的regular expression而已，複雜的regular expression真的是給鬼看的，可讀性不是你出一張嘴說了就算，可讀性根本就和讀者的能力扯不上邊，請問你寫了一個只有你自己看得懂的程式，你說你很強，別人看不懂是程度爛，你的程式叫可讀性很好嗎? 是可笑吧? 我是老闆的話肯定開除這種人

我說過了，你要麻就也寫一個amazon.com的regular expression的版本我們來比較看看，如果你真的覺得amazon.com太簡單太整齊，要不然你也挑一個網站當題目? 出一張嘴誰都會，打嘴砲我也沒輸過，出一張嘴就想潑別人冷水?

---

Joe *says:*
2008 年 12 月 08 日 at 1:44 下午

你贏了可以吧
又不想弄個程式比較來比較去
regular expression可以抓amazon.com的資料又不是網路上查不到的事情
我貼上程式碼又能改變你什麼嗎？反正你就認定regular expression不好讀嗎….

反正你就認定是鬼畫符，這個要怎麼改變？

---

Joe *says:*
2008 年 12 月 08 日 at 1:49 下午

xpath，你叫他切片法？

摸摸你的良心？

victor *says:*
2008 年 12 月 08 日 at 1:57 下午

首先是你說切片可讀性不佳，我就說了那是你誤會，而且我說過適合的時機用適合的工具，你也認定切片不好讀，我都說那是誤解，也提出證據來說明，regular expression本來就不好讀，而你呢? 你要說我說的是不對的，我有說不接受嗎? 你好歹也給點資料參考好嗎? 出一張嘴誰都會

如果你覺得amazon.com太麻煩，要不然你用regular expression改寫我那個抓tw.yahoo.com的程式好嗎? 如果你覺得在我的blog都是我在說，那也可以找一個PTT的討論版或哪裡我們貼上去看大家怎麼認為

但你從頭到尾就出一張嘴，也什麼都沒給，"反正網路上又不是查不到"，我該說什麼? 這不是打嘴砲是什麼? 相左的意見我很歡迎也會尊重，但是像這樣只出一張嘴，文章也不仔細看，也不給資料就要打嘴砲，我能說什麼?

xpath我哪裡說過它是切片= =|| 我該說什麼呢? 你文章不仔細看就要戰，我也沒辦法阿 ..╮(￣﹏￣)╭..

victor *says:*
2008 年 12 月 08 日 at 1:59 下午

我都說在使用regular的前面還有lxml和BeautifulSoup在做parse的工作，解析成tree後如果文字還有格式才需要切片，我從頭到尾到底那裡有說過xpath等於切片??? 你可以找出來給我看嗎? 自己文章看不仔細又愛戰，我也無言了

Joe *says:*
2008 年 12 月 08 日 at 2:03 下午

一開始就說 我討論的廣泛程式，不屑的是切片法
現在你跟我說 沒在討論切片法，算你厲害……
google會不會？
http://www.google.com.tw/search?hl=zh-TW&rlz=1C1GGLS_zh-TWTW294TW303&q=amazon+source+code+regular&btnG=搜尋&meta=&aq=f&oq=

Joe *says:*
2008 年 12 月 08 日 at 2:05 下午

為什麼你的regular 前面還要 做parse的工作？
你不會的東西很多……..

victor *says:*
2008 年 12 月 08 日 at 8:27 下午

什麼叫算我厲害= = 你比較厲害吧，從頭倒尾我到底說過幾次你誤會了，叫你重看文章多少次，你到底看了嗎? 根本就沒看仔細還可以戰那麼，要跟人戰還有放大絕你不會自己Google嗎? 要不要你老闆要你寫程式你也跟他說你不會自己Google嗎? 你好歹也貼個頁面連結給我看，貼那垃圾連結誰不會，還不是一

樣在打嘴砲，一點進步都沒有

regular前面為什麼做parse，不然怎麼用xpath? 你不自己都看見xpath了嗎? 不然你以為xpath是免錢的嗎?

真的受不了，天兵成這樣還要跟人戰= ="

---

DianQ *says:*

2009 年 03 月 03 日 at 10:36 下午

victor 我从python邮件列表 跑来拜读你的文章，写的真不错 准备入手学习一下 Twisted + lxml

---

yookoyoo *says:*

2010 年 03 月 25 日 at 11:08 上午

你写得不错，非常好！

---

guest *says:*

2010 年 08 月 25 日 at 12:24 下午

那個joe根本是xx…
一直在離題 也說不出個所以然 越講越遠
講輸了就再給你牽脫別的東西出來
這種人在程式界還不少…orz

在發表抱怨文同時 你的文章也正在被大家檢視
在我看你 這場嘴砲之爭是joe輸了
因為我根本不知道他在扯什麼東西

---

Sophia *says:*

2010 年 09 月 15 日 at 8:05 下午

之前寫過C++,Java,ASP, 後來有一段時間都在做資料分析相關的工作,所以都寫SQL去處理資料,最近想要知道如果抓網站上股票相關資料來計算自己想要參考的報表,看到您的作品裡有這個部份,懇請您賜教, 謝謝!!

看您的說明後,決定花點時間研究一下 Python…

---

GourryMK2 *says:*

2010 年 11 月 20 日 at 2:42 上午

PHP 下有個叫 phpquery 的函式庫, 它完整地重現了 jquery 處理 DOM 的方式, 在解析網頁上面非常地好用, 因為利用 DOM 存取的方式可以省掉很多複雜的 Regular Expression, 不知道 python 是否有類似的函式庫? 還是大家都是只靠 Regular Expression ?

---

路人假 *says:*

2011 年 04 月 27 日 at 5:04 下午

看完下面JOE vs. Victor 終於懂了
沒有最佳程式
只有最適合程式

但適合不適合就見仁見智了 ^_^

XianYeeXing *says:*
2011 年 05 月 22 日 at 11:19 上午

Mr.:
thank u for there great shares !

now u got another option : nodeJS !

Leonard *says:*
2011 年 08 月 24 日 at 4:56 下午

不知您覺得Scrapy這個函式庫用起來如何呢？最近準備用它來抓網頁，它似乎把抓網頁該有的功能都寫好了，而且可以自行指定深度呢…

liangguohuan *says:*
2011 年 12 月 04 日 at 6:33 下午

麻烦帮我看下面代码,怎么打印出来的东西变成了十六进制,调用格式化函数document_fromstring()后里边转换了,如果用lxml.html.soupparser.fromstring()转换就不会出问题,怎么解决啊!!
# -*- coding: utf-8 -*-
"‘
Created on 2011-12-3

@author: hanson
"‘
from lxml import etree
import urllib2
def quick_parse():
url = "http://blog.ez2learn.com/2008/10/05/python-is-the-best-choice-to-grab-web/"
strhtml = urllib2.urlopen(url).read()
#f = StringIO(strhtml)
parser = etree.HTMLParser()
tree = etree.fromstring(strhtml,parser)
#tree = lxml.html.document_fromstring(strhtml,parser)
for node in tree.xpath(‘//ul[@class="xoxo"]//ul[last()]/li/a’):
#print node.text
print repr(node.text)
quick_parse()

## 發表迴響

您的電子郵件位址並不會被公開。 必要欄位標記為 *

名稱 *

電子郵件 *

個人網站

迴響

您可以使用這些 HTML 標籤與屬性： <a href="" title=""> <abbr title=""> <acronym title="">
<b> <blockquote cite=""> <cite> <code> <del datetime="">  <i> <q cite="">
<strike>  <pre lang="" line="" escaped="" highlight="">