Python requests

11/18

Installation

```
pip install requests
```

Using requests

```
import requests
```

GET

伺服器的回應

```
res = request.get(url)
```

跟伺服器請求資源

Response Object

dir(res)

查看 response object 有哪些 attribute

```
['__attrs__', '__bool__', '__class__', '__delattr__', '__dict__', '__dir__',
'__doc__', '__enter__', '__eq__', '__exit__', '__format__', '__ge__',
'__getattribute__', '__getstate__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__lt__', '__module__', '__ne__',
'__new__', '__nonzero__', '__reduce__', '__reduce_ex__', '__repr__',
'__setattr__', '__setstate__', '__sizeof__', '__str__', '__subclasshook__',
'__weakref__', '_content', '_content_consumed', '_next',
'apparent_encoding', 'close', 'connection', 'content', 'cookies',
'elapsed', 'encoding', 'headers', 'history', 'is_permanent_redirect',
'is_redirect', 'iter_content', 'iter_lines', 'json', 'links', 'next', 'ok',
'raise_for_status', 'raw', 'reason', 'request', 'status_code', 'text', 'url']
```

Response Object

res.status_code

回傳網頁
status
code

Response Object

res.text

回傳網頁 html

Response Object

res.encoding

To display an HTML page correctly, a web browser must know which character set to use.

UTF-8

ISO-8859-1

ASCII

Response Object

res.cookie

Response Object

res.header

HTTP headers let the client and the server pass additional information with an HTTP request or response. An HTTP header consists of its case-insensitive name followed by a colon (`:`), then by its value. Whitespace before the value is ignored.

靜態網頁爬蟲

用途：

1. 研究需要自己爬資料

2. 自動化，不用一直重複輸入網址

3. …

靜態網頁爬蟲

如果 **res.text** 可以直接翻譯成網頁上的資訊（觀察 **html tags**），基本上就可以用靜態爬蟲。

觀察 url

如果有 **query string** 出現，代表我們可以直接在 **url** 裡面塞 **query string**，就可以輕易地用 **Python** 發 **request**。

Caveats

如果有登入就比較麻煩，無法直接用 **requests**。

觀察 url

https:/ntuee.org?id=b06901104

如果網頁出現這個代表我們可以直接操作 id 這個 query parameter。

Requets 傳參數

- 直接放在 url 裡面
- 如果很多 parameters，可以用 params。

```
res = request.get(url, params = ...)
```

解析 html with bs4

- bs4 (BeautifulSoup) 就像一個 html parser
- 可以把 res.text 解析成易於操作的形式，然後我們可以用 bs4 內建的函式去存取 html 內容。

BeautifulSoup

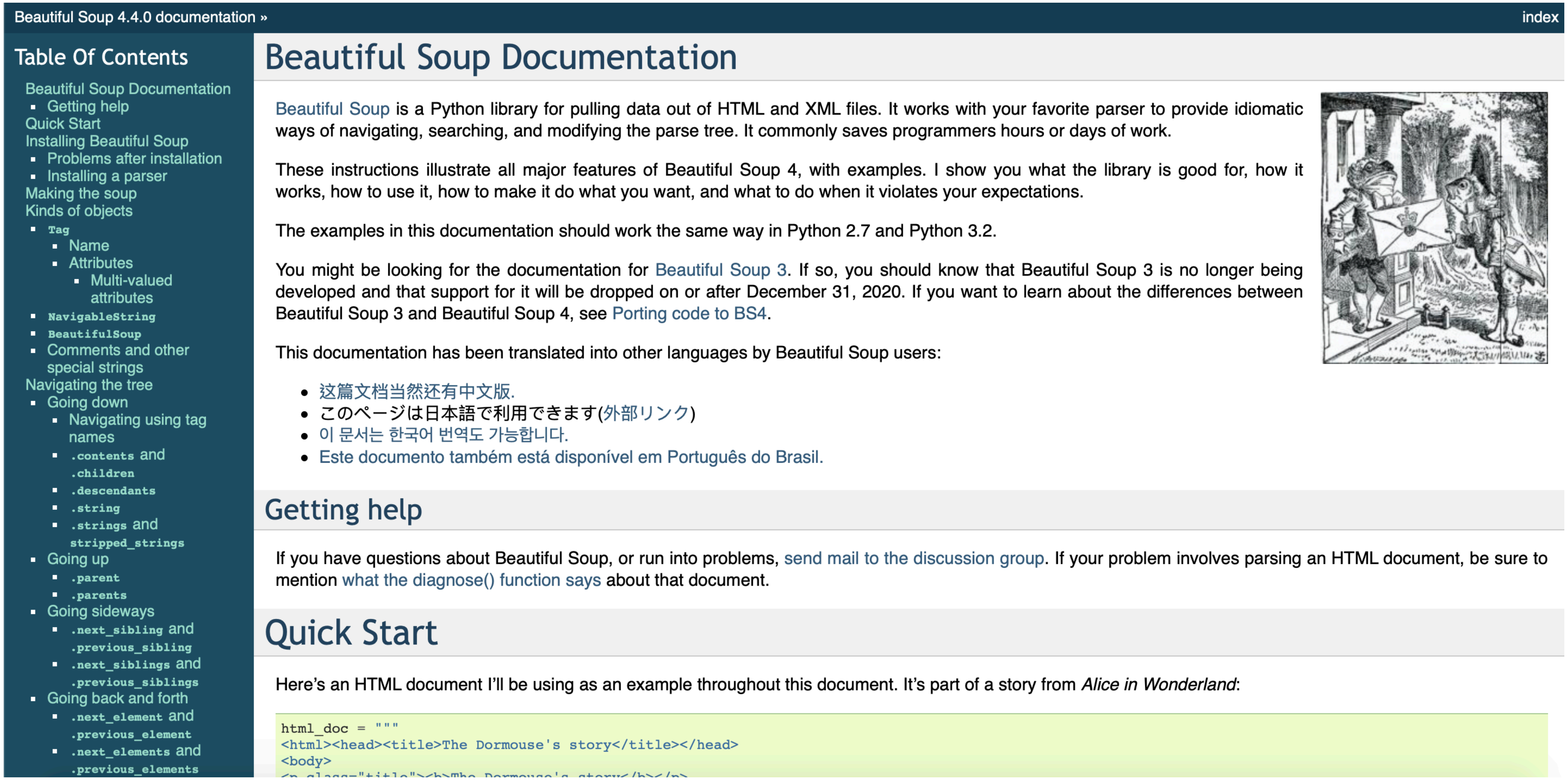
```
from bs4 import BeautifulSoup

soup = BeautifulSoup(res.text, ‘html.parser’)
```

接下來就可以對 soup 物件進行操作！

BeautifulSoup

- 請愛用 Documentation



Beautiful Soup 4.4.0 documentation »

index

Beautiful Soup Documentation

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

These instructions illustrate all major features of Beautiful Soup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

The examples in this documentation should work the same way in Python 2.7 and Python 3.2.

You might be looking for the documentation for Beautiful Soup 3. If so, you should know that Beautiful Soup 3 is no longer being developed and that support for it will be dropped on or after December 31, 2020. If you want to learn about the differences between Beautiful Soup 3 and Beautiful Soup 4, see Porting code to BS4.

This documentation has been translated into other languages by Beautiful Soup users:

- 这篇文档当然还有中文版.
- このページは日本語で利用できます(外部リンク)
- 이 문서는 한국어 번역도 가능합니다.
- Este documento também está disponível em Português do Brasil.

Getting help

If you have questions about Beautiful Soup, or run into problems, send mail to the discussion group. If your problem involves parsing an HTML document, be sure to mention what the diagnose() function says about that document.

Quick Start

Here's an HTML document I'll be using as an example throughout this document. It's part of a story from *Alice in Wonderland*:

```
html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>
```

今日任務：

找到所有電機系教授的信箱

其他有用的 library

urllib