Beautiful Soup 4 Python

Overview

```
This article is an introduction to BeautifulSoup 4 in Python.

If you want to know more I recommend you to read the official documentation
found here (http://www.crummy.com/software/BeautifulSoup/bs4/doc/).
```

What is Beautiful Soup?

```
Beautiful Soup is a Python library for pulling data out of HTML and XML files.
```

BeautifulSoup 3 or 4?

```
Beautiful Soup 3 has been replaced by Beautiful Soup 4.

Beautiful Soup 3 only works on Python 2.x, but Beautiful Soup 4 also works on
Python 3.x.

Beautiful Soup 4 is faster, has more features, and works with third-party parsers
like lxml and html5lib.

You should use Beautiful Soup 4 for all new projects.
```

Installing Beautiful Soup

```
If you run Debian or Ubuntu, you can install Beautiful Soup with the system
package manager
```

```
apt-get install python-bs4
```

```
Beautiful Soup 4 is published through PyPi, so if you can’t install it with the
system packager, you can install it with easy_install or pip.

The package name is beautifulsoup4, and the same package works on Python 2 and
Python 3.
```

```
easy_install beautifulsoup4

pip install beautifulsoup4
```

```
If you don’t have easy_install or pip installed, you can download (http://www.crummy.c
om/software/BeautifulSoup/bs4/download/4.0/) the Beautiful
Soup 4 source tarball and install it with setup.py.

python setup.py install
```

BeautifulSoup Usage

```
Right after the installation you can start using BeautifulSoup.

At the beginning of your Python script, import the library

Now you have to pass something to BeautifulSoup to create a soup object.

That could be a document or an URL.

BeautifulSoup does not fetch the web page for you, you have to do that yourself.

That's why I use urllib2 in combination with the BeautifulSoup library.
```

Filtering

```
There are some different filters you can use with the search API.

Below I will show you some examples on how you can pass those filters into
methods such as find_all

You can use these filters based on a tag’s name, on its attributes, on the text
of a string, or on some combination of these.
```

A string

```
The simplest filter is a string.

Pass a string to a search method and Beautiful Soup will perform a match against
that exact string.

This code finds all the 'b' tags in the document (you can replace b with any
tag you want to find)
```

```
soup.find_all('b')
```

```
If you pass in a byte string, Beautiful Soup will assume the string is encoded
as UTF-8.

You can avoid this by passing in a Unicode string instead.
```

A regular expression

```
If you pass in a regular expression object, Beautiful Soup will filter against
that regular expression using its match() method.

This code finds all the tags whose names start with the letter "b",
in this case, the 'body' tag and the 'b' tag:
```

```
import re
for tag in soup.find_all(re.compile("^b")):
    print(tag.name)
```

```
This code finds all the tags whose names contain the letter "t":
```

```
for tag in soup.find_all(re.compile("t")):
    print(tag.name)
```

A list

```
If you pass in a list, Beautiful Soup will allow a string match against any
item in that list.

This code finds all the 'a' tags and all the 'b' tags
```

```
print soup.find_all(["a", "b"])
```

True

```
The value True matches everything it can.

This code finds all the tags in the document, but none of the text strings:
```

```
for tag in soup.find_all(True):
    print(tag.name)
```

A function

```
If none of the other matches work for you, define a function that takes an
element as its only argument.

Please see the official documentation if you want to do that.
```

BeautifulSoup Object

```
As an example, we'll use the very website you currently are on
(https://www.pythonforbeginners.com)

To parse the data from the content, we simply create a BeautifulSoup object for it

That will create a soup object of the content of the url we passed in.

From this point, we can now use the Beautiful Soup methods on that soup object.

We can use the prettify method to turn a BS parse tree into a nicely formatted
Unicode string
```

The Find_all method

```
The find_all method is one of the most common methods in BeautifulSoup.

It looks through a tag’s descendants and retrieves all descendants that match
your filters.
```

```
soup.find_all("title")

soup.find_all("p", "title")

soup.find_all("a")

soup.find_all(id="link2")
```

```
Let's see some examples on how to use BS 4
```

```
from bs4 import BeautifulSoup
import urllib2

url = "https://www.pythonforbeginners.com"

content = urllib2.urlopen(url).read()

soup = BeautifulSoup(content)

print soup.prettify()

print title
>> 'title'? Python For Beginners

print soup.title.string
>> ? Python For Beginners

print soup.p




print soup.a
Python For Beginners (https://www.pythonforbeginners.com)
```

Navigating the Parse Tree

```
If you want to know how to navigate the tree please see the official documentation (ht
tp://www.crummy.com/software/BeautifulSoup/bs4/doc/#navigating-the-tree)

There you can read about the following things:

Going down
        Navigating using tag names
        .contents and .children
        .descendants
        .string
        .strings and stripped_strings

Going up
        .parent
        .parents

Going sideways
        .next_sibling and .previous_sibling
        .next_siblings and .previous_siblings

Going back and forth
        .next_element and .previous_element
        .next_elements and .previous_elements
```

Extracting all the URLs found within a page 'a' tags

```
One common task is extracting all the URLs found within a page’s 'a' tags

Using the find_all method, gives us a whole list of elements with the tag "a".
```

```
for link in soup.find_all('a'):
    print(link.get('href'))
```

```
Output:
```

```
..https://www.pythonforbeginners.com
..https://www.pythonforbeginners.com/python-overview-start-here/
..https://www.pythonforbeginners.com/dictionary/
..https://www.pythonforbeginners.com/python-functions-cheat-sheet/
..https://www.pythonforbeginners.com/lists/python-lists-cheat-sheet/
..https://www.pythonforbeginners.com/loops/
..https://www.pythonforbeginners.com/python-modules/
..https://www.pythonforbeginners.com/strings/
..https://www.pythonforbeginners.com/sitemap/
...
...
```

Extracting all the text from a page

```
Another common task is extracting all the text from a page:
```

```
print(soup.get_text())
```

```
Output:
```

```
Python For Beginners
Python Basics
Dictionary
Functions
Lists
Loops
Modules
Strings
Sitemap
...
...
```

Get all links from Reddit

```
As a last example, let's grab all the links from Reddit
```

```
from bs4 import BeautifulSoup
import urllib2

redditFile = urllib2.urlopen("http://www.reddit.com")
redditHtml = redditFile.read()
redditFile.close()

soup = BeautifulSoup(redditHtml)
redditAll = soup.find_all("a")
for links in soup.find_all('a'):
    print (links.get('href'))
```

```
Output:
```

```
#content
..http://www.reddit.com/r/AdviceAnimals/
..http://www.reddit.com/r/announcements/
..http://www.reddit.com/r/AskReddit/
..http://www.reddit.com/r/atheism/
..http://www.reddit.com/r/aww/
..http://www.reddit.com/r/bestof/
..http://www.reddit.com/r/blog/
..http://www.reddit.com/r/funny/
..http://www.reddit.com/r/gaming/
..http://www.reddit.com/r/IAmA/
..http://www.reddit.com/r/movies/
..http://www.reddit.com/r/Music/
..http://www.reddit.com/r/pics/
..http://www.reddit.com/r/politics/
...
```

```
For more information, please see the official (https://beautiful-soup-4.readthedocs.or
g/en/latest/) documentation.
```

Read more about: