The `BeautifulSoup` Package

> Prerequisite: the `requests` package

The `BeautifulSoup` package provides a simple way of parsing a website's HTML contents.

Reference: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

Installation

First install the package, if necessary:

```
pip install beautifulsoup4 # note the 4 at the end - this is the latest
version
```

Usage

You can use this package from the command line or from within a script. The examples below depict usage from within a script.

When you make a request that returns some HTML string, you can parse it like this:

```
import requests
from bs4 import BeautifulSoup # note that the import package command is `bs4`

response = requests.get("https://www.gutenberg.org/ebooks/author/65")
response_html = response.text

soup = BeautifulSoup(response_html)

titles = soup.find_all("span", "title")

print(type(titles)) #> <class 'bs4.element.ResultSet'> (like a list)
print(titles[5]) #> <span class="title">Macbeth</span>
print(titles[5].text) #> Macbeth

booklinks = soup.find_all("li", "booklink")

books = []
for list_item in booklinks:
    title = list_item.find("span", "title").text #> "Shakespeare's Sonnets"
    author = list_item.find("span", "subtitle").text #> "William Shakespeare"
    downloads = list_item.find("span", "extra").text #> '830 downloads'
    downloads_count = int(downloads.replace(" downloads", "")) #> 830
```

```
    book = {"title": title, "author": author, "downloads": downloads_count}
    print(book)
    books.append(book)

print(books[2]["title"]) #> Macbeth
```

It's easier to parse HTML once you know the document structure. Try using your browser's developer tools to examine the document structure of any web page. For example, in Google Chrome you can right-click on a webpage and select "Inspect".