**beautifulsoup+json抓取stackoverflow**

本文用beautifulsoup4库抓取stackoverflow上python最新问题，存储到json文件。前半部分通过抓取多个字段练习beautifulsoup的使用，后半部分介绍json模块

关于beautifulsoup的使用可以参考这篇文章[BeautifulSoup全面总结](https://link.juejin.im/?target=https%3A%2F%2Fzhuanlan.zhihu.com%2Fp%2F35354532" \t "_blank)

**爬虫代码**

import requests # 导入网页请求库

from bs4 import BeautifulSoup # 导入网页解析库

import re

import json

class Stack(object):

def \_\_init\_\_(self):

self.baseurl = 'https://stackoverflow.com' # 用于与抓取的url拼接

self.starturl = 'https://stackoverflow.com/questions/tagged/python' # 初始url

def start\_requests(self, url): # 发起请求

r = requests.get(url)

return r.content

def parse(self, text): # 解析网页

soup = BeautifulSoup(text, 'html.parser')

divs = soup.find\_all('div', class\_ = 'question-summary')

for div in divs:

# 一些中间变量

gold = div.find('span', title = re.compile('gold'))

silver = div.find('span', title = re.compile('silver'))

bronze = div.find('span', title = re.compile('bronze'))

tags = div.find('div', class\_ = 'summary').find\_all('div')[1].find\_all('a')

# 用生成器输出字典

yield {

# 这部分每一条都有代表性

'title': div.h3.a.text,

'url': self.baseurl + div.h3.a.get('href'),

'answer': div.find('div', class\_ = re.compile('status')).strong.text,

'view': div.find('div', class\_ = 'views ').text[: -7].strip(),

'gold': gold.find('span', class\_ = 'badgecount').text if gold else 0,

'tagnames': [tag.text for tag in tags],

# 下面的从知识的角度上讲都和上面一样

'vote': div.find('span', class\_ = 'vote-count-post ').strong.text,

'time': div.find('div', class\_ = 'user-action-time').span.get('title'),

'duration': div.find('div', class\_ = 'user-action-time').span.text,

'username': div.find('div', class\_ = 'user-details').a.text,

'userurl': self.baseurl + div.find('div', class\_ = 'user-gravatar32').a.get('href'),

'reputation': div.find('span', class\_ = 'reputation-score').text,

'silver': silver.find('span', class\_ = 'badgecount').text if silver else 0,

'bronze': bronze.find('span', class\_ = 'badgecount').text if bronze else 0,

'tagurls': [self.baseurl + tag.get('href') for tag in tags]

}

# 启动爬虫

def start(self):

text = self.start\_requests(self.starturl)

items = self.parse(text)

s = json.dumps(list(items), indent = 4, ensure\_ascii=False)

with open('stackoverflow.json', 'w', encoding = 'utf-8') as f:

f.write(s)

stack = Stack()

stack.start()

**json模块介绍**

json是一个内置模块，无需自己安装，模块主要就用两个函数json.dumps和json.loads

* 前者可以把一个list dict的python对象变成样子相同的字符串，这样转化一般用于存储到json文件中，因为json文件的形式和list dict是一样的，而存储文件需要使用字符串（或者bytes）
* 后者将list dict样子的字符串转化为python对象，如果读取json文件，得到的就是这样的字符串，通过这个转化将其变成python可以处理的list dict

示例代码展示如下

import json

a = [{'name':'Bob', 'age': 20}, {'name': 'Mary', 'age': 18}]

s = json.dumps(a)

s # 一个字符串

# '[{"age": 20, "name": "Bob"}, {"age": 18, "name": "Mary"}]'

b = json.loads(s)

b[0]

# {'age': 20, 'name': 'Bob'}

b[0].get('age')

# 20

存储到文件时，为了让字符串展示更好看一些，还有编码问题，一般加参数如下

存储到文件

s = json.dumps(a, indent = 4, ensure\_ascii=False)

with open('a.json', 'w', encoding = 'utf-8') as f:

f.write(s)

参数indent指定一些缩进，不然写到文件里所有字符都堆在一起不方便看。

ensure\_ascii则是存储内容涉及中文时需要指定（上面抓取stackoverflow没有中文，所以其实是不需要指定的，只是为了引出这个参数才这么用）

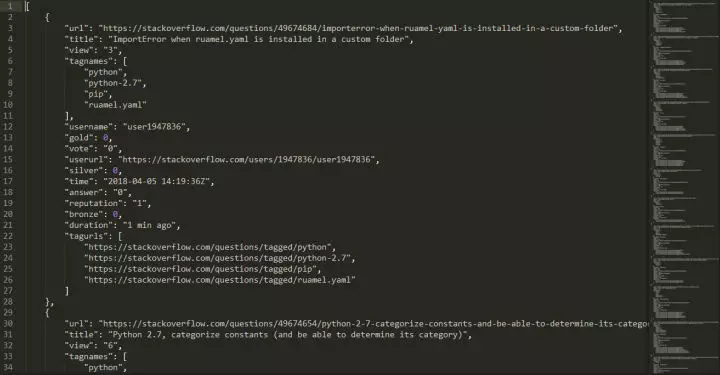
从文件中读取

with open('a.json', encoding = 'utf-8') as f:

s = f.read()

b = json.loads(s)

抓取结果如下图所示



# 对爬取数据进行简单分析

## votes 分析

### 降序排列了 votes 数，生成折线图

Votes折线图

2k 后的问题的 votes 数基本上就已经在 400 以下了，接着后面的就基本上是贴地飞行了  
votes 数最多 : [Why is it faster to process a sorted array than an unsorted array?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/11227809/why-is-it-faster-to-process-a-sorted-array-than-an-unsorted-array)

### votes 数的连续分布情况

votes甘特图

可见最多的还是集中在 1-2K 之间,从 6k 开始基本上就断层了

### 具体数据

| **description** | **count** |
| --- | --- |
| votes >= 500 | 1630 |
| votes >= 400 | 2325 |
| votes >= 300 | 3782 |
| votes >= 200 | 7062 |
| votes >= 100 | 19781 |

如果以 100 为分界线的话，会得到这样的一个饼图

pie\_votes\_1

大于 100 的连 %2 都不到

再来看看底层的数据

| **description** | **count** |
| --- | --- |
| 1 <= votes <= 5 | 211804 |
| 6 <= votes <= 10 | 430935 |
| 11 <= votes <= 15 | 136647 |
| 16 <= votes <= 20 | 64541 |
| votes <= 20 | 843927 |

可见 votes 小于 20 的，数量高达 84m  
看看总体的比例吧

pie\_votes\_2

## answers 分析

### 降序排列了 answers 数，生成折线图

answers折线图

很明显 3k 之后的 answers 数基本上就小于 20 了  
answers 数最多: [What is the best comment in source code you have ever encountered? [closed]](https://link.jianshu.com/?t=http://stackoverflow.com/questions/184618/what-is-the-best-comment-in-source-code-you-have-ever-encountered)

### answers 数的连续分布情况

answers甘特图

150 后也就断层了，实际上能达到这样的回答数极少

### 具体数据

| **description** | **count** |
| --- | --- |
| answers >= 5 | 218059 |
| answers >= 10 | 34500 |
| answers >= 20 | 3808 |
| answers >= 30 | 968 |

大于 30 的确实少的可怜，看看总体情况

pie\_answer\_1

## views 分析

### 降序排列了 views 数，生成折线图

views折线图

最高达到了 4.5m，100000 以后的基本上就不足 28000 了  
views 数最多: [How to undo last commit(s) in Git?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/927358/how-to-undo-last-commits-in-git)

### views 数的连续分布情况

views甘特图

### 具体数据

| **description** | **count** |
| --- | --- |
| views >= 5000 | 486466 |
| views >= 10000 | 315576 |
| views >= 20000 | 171873 |
| views >= 50000 | 59363 |
| views >= 100000 | 22224 |
| views >= 200000 | 7030 |

大部分问答的 views 数还是集中在 20000 以内  
还是得看看总体分布

bubble\_views

## 再看看 votes，views，answers 三者的散点图对应情况

### votes - views

votes-views散点图

### votes - answers

votes-answers散点图

### views - answers

views-answers散点图

总的来说，这三者对应关系类似于一个金字塔。三个图基本上都是左下角靠近原点的区域被填满，也就是说绝对大部分的问题的 votes，answers 和 views 都是属于最下层的。高质量活跃的问题是处于金字塔顶端的。三者的最高数好像也没特别明显的对应关系，且三者的最高数都不是同一个问题。

根据所有问题的 tags 提取出总量前 200 的关键词（前 50 条如下），第 1 名是 c#，python 排在第 5

('c#', 94614),

('java', 93244),

('javascript', 76722),

('android', 69321),

('python', 62502),

('c++', 58173),

('php', 42596),

('ios', 37773),

('jquery', 37405),

('.net', 36180),

('html', 28536),

('css', 26174),

('c', 24699),

('objective-c', 23253),

('iphone', 22171),

('ruby-on-rails', 20143),

('sql', 19171),

('asp.net', 18060),

('mysql', 17559),

('ruby', 16397),

('r', 15670),

('git', 13139),

('linux', 13080),

('asp.net-mvc', 12857),

('angularjs', 12606),

('sql-server', 12473),

('node.js', 12212),

('django', 11576),

('arrays', 11006),

('algorithm', 10959),

('wpf', 10631),

('performance', 10619),

('xcode', 10613),

('string', 10426),

('windows', 10132),

('eclipse', 10117),

('scala', 9942),

('regex', 9685),

('multithreading', 9601),

('json', 9266),

('swift', 8950),

('c++11', 8939),

('haskell', 8823),

('osx', 8159),

('visual-studio', 8140),

('html5', 7627),

('database', 7567),

('xml', 7478),

('spring', 7464),

('unit-testing', 7253),

('bash', 6825)

## Python 类的问答

### votes 数前 10

* 6162 : [What does the “yield” keyword do in Python?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/231767/what-does-the-yield-keyword-do-in-python)
* 3529 : [What is a metaclass in Python?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/100003/what-is-a-metaclass-in-python)
* 3098 : [How do I check whether a file exists using Python?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/82831/how-do-i-check-whether-a-file-exists-using-python)
* 3035 : [Does Python have a ternary conditional operator?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/394809/does-python-have-a-ternary-conditional-operator)
* 2620 : [Calling an external command in Python](https://link.jianshu.com/?t=http://stackoverflow.com/questions/89228/calling-an-external-command-in-python)
* 2605 : [What does if **name** == “**main**”: do?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/419163/what-does-if-name-main-do)
* 2194 : [How to merge two Python dictionaries in a single expression?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/38987/how-to-merge-two-python-dictionaries-in-a-single-expression)
* 2123 : [Sort a Python dictionary by value](https://link.jianshu.com/?t=http://stackoverflow.com/questions/613183/sort-a-python-dictionary-by-value)
* 2058 : [How to make a chain of function decorators?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/739654/how-to-make-a-chain-of-function-decorators)
* 1984 : [How to check if a directory exists and create it if necessary?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/273192/how-to-check-if-a-directory-exists-and-create-it-if-necessary)

### answers 数前 10

* 191 : [Hidden features of Python [closed]](https://link.jianshu.com/?t=http://stackoverflow.com/questions/101268/hidden-features-of-python)
* 87 : [Best ways to teach a beginner to program? [closed]](https://link.jianshu.com/?t=http://stackoverflow.com/questions/3088/best-ways-to-teach-a-beginner-to-program)
* 55 : [Favorite Django Tips & Features?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/550632/favorite-django-tips-features)
* 50 : [How do you split a list into evenly sized chunks?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/312443/how-do-you-split-a-list-into-evenly-sized-chunks)
* 44 : [Calling an external command in Python](https://link.jianshu.com/?t=http://stackoverflow.com/questions/89228/calling-an-external-command-in-python)
* 43 : [How can I represent an 'Enum' in Python?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/36932/how-can-i-represent-an-enum-in-python)
* 38 : [How to merge two Python dictionaries in a single expressions](https://link.jianshu.com/?t=http://stackoverflow.com/questions/38987/how-to-merge-two-python-dictionaries-in-a-single-expression)
* 38 : [Finding local IP addresses using Python's stdlib](https://link.jianshu.com/?t=http://stackoverflow.com/questions/166506/finding-local-ip-addresses-using-pythons-stdlib)
* 37 : [Reverse a string in python without using reversed or [::-1]](https://link.jianshu.com/?t=http://stackoverflow.com/questions/18686860/reverse-a-string-in-python-without-using-reversed-or-1)
* 37 : [How do I check whether a file exists using Python?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/82831/how-do-i-check-whether-a-file-exists-using-python)

### views 数前 10

* 2121621 : [Parse String to Float or Int](https://link.jianshu.com/?t=http://stackoverflow.com/questions/379906/parse-string-to-float-or-int)
* 1905938 : [Using global variables in a function other than the one that created them](https://link.jianshu.com/?t=http://stackoverflow.com/questions/423379/using-global-variables-in-a-function-other-than-the-one-that-created-them)
* 1888666 : [How do I check whether a file exists using Python?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/82831/how-do-i-check-whether-a-file-exists-using-python)
* 1827126 : [Calling an external command in Python](https://link.jianshu.com/?t=http://stackoverflow.com/questions/89228/calling-an-external-command-in-python)
* 1699574 : [Converting integer to string in Python?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/961632/converting-integer-to-string-in-python)
* 1686230 : [How do I read a file line-by-line into a list?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/3277503/how-do-i-read-a-file-line-by-line-into-a-list)
* 1682307 : [Iterating over dictionaries using 'for' loops in Python](https://link.jianshu.com/?t=http://stackoverflow.com/questions/3294889/iterating-over-dictionaries-using-for-loops-in-python)
* 1569205 : [How to get the size of a list](https://link.jianshu.com/?t=http://stackoverflow.com/questions/1712227/how-to-get-the-size-of-a-list)
* 1554755 : [How do I install pip on Windows?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/4750806/how-do-i-install-pip-on-windows)
* 1515505 : [Finding the index of an item given a list containing it in Python](https://link.jianshu.com/?t=http://stackoverflow.com/questions/176918/finding-the-index-of-an-item-given-a-list-containing-it-in-python)

### 三者的前 10 中有 2 个问题是完全重叠的，分别是

* [How do I check whether a file exists using Python?](https://link.jianshu.com/?t=http://stackoverflow.com/questions/82831/how-do-i-check-whether-a-file-exists-using-python)
* [Calling an external command in Python](https://link.jianshu.com/?t=http://stackoverflow.com/questions/89228/calling-an-external-command-in-python)