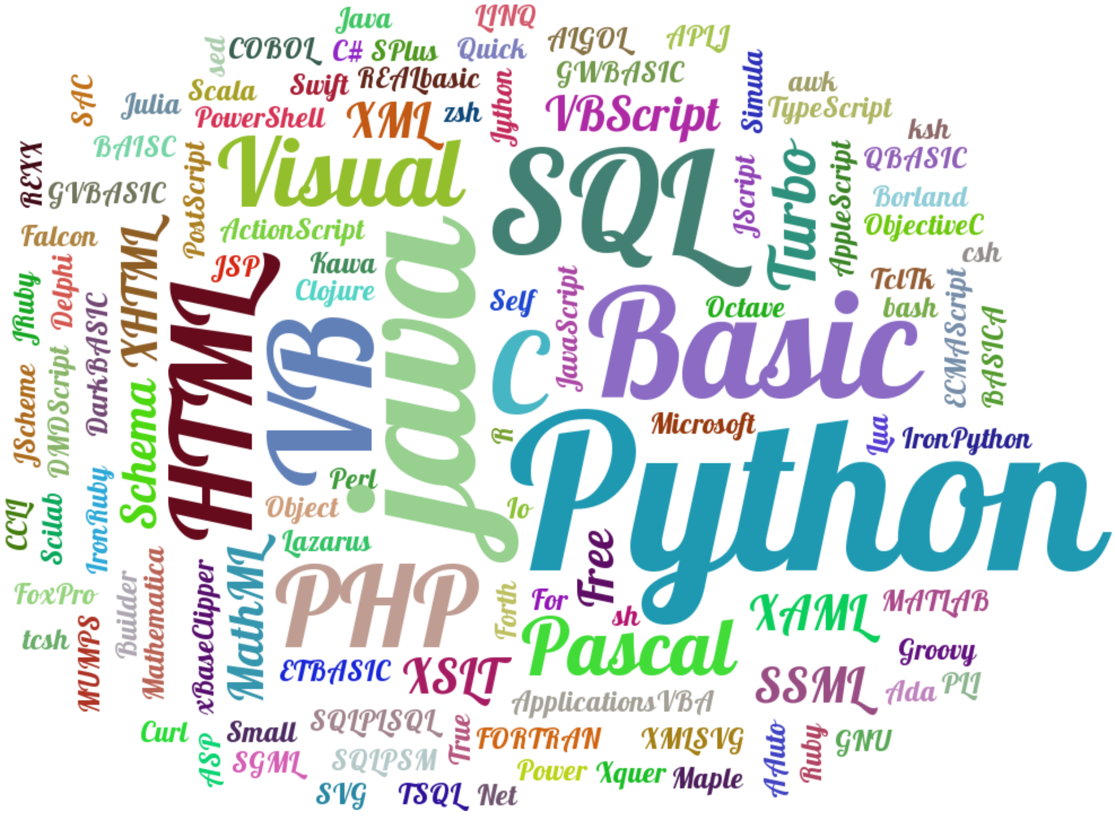
标签云是现在大数据里面最喜欢使用的一种展现方式，其中在 python3 下也能实现标签云的效果，贴图如下：



-------------------进入正文---------------------

首先要安装以下几个库：

```
#!/usr/bin/python3.4
# -*- coding: utf-8 -*-

# http://www.lfd.uci.edu/~gohlke/pythonlibs/#cx_freeze
# 万能仓库下载 pygame
# pip3 下载 simplejson
```

还有最重要的库：

```
pip3 install pytagcloud
```

或者去官网下载：

```
https://pypi.python.org/pypi/pytagcloud/
```

安装完毕，利用官网的例子来做：

```
from pytagcloud import create_tag_image, make_tags
from pytagcloud.lang.counter import get_tag_counts

```

```
YOUR_TEXT = "A tag cloud is a visual representation for text data, typically\
used to depict keyword metadata on websites, or to visualize free form text."

tags = make_tags(get_tag_counts(YOUR_TEXT), maxsize=120)

create_tag_image(tags, 'cloud_large.png', size=(900, 600), fontname='Lobster')
```

果断报错：

```
Traceback (most recent call last):
  File "D:/code/pythonwork/Text.py", line 96, in <module>
    tags = make_tags(get_tag_counts(YOUR_TEXT), maxsize=120)
  File "C:\Python34\lib\site-packages\pytagcloud\lang\counter.py", line 25, in get_tag_counts
    return sorted(counted.iteritems(), key=itemgetter(1), reverse=True)
AttributeError: 'dict' object has no attribute 'iteritems'
```

看了发现问题出在库中的：

```
# counter.py
return sorted(counted.iteritems(), key=itemgetter(1), reverse=True)
```

原来是 python3.4 不支持写法：

在 Python2.x 中，items( )用于 返回一个字典的拷贝列表【Returns a copy of the list of all items (key/value pairs) in D】，占额外的内存。

iteritems() 用于返回本身字典列表操作后的迭代【Returns an iterator on all items(key/value pairs) in D】，不占用额外的内存。

Python 3.x 里面，iteritems() 和 viewitems() 这两个方法都已经废除了，而 items() 得到的结果是和 2.x 里面 viewitems() 一致的。在 3.x 里 用 items()替换 iteritems() ，可以用于 for 来循环遍历。

但是当我换成：

```
# counter.py
return sorted(counted.items(), key=itemgetter(1), reverse=True)
```

发现运行并没有错误，但是没有生成标签云啊，一遍一遍打印出来，终于找到问题了：

```
from pytagcloud import create_tag_image
```

这个是为了生成一个元组的东西：

```
    # counts =[('cloud', 3),
    # ('words', 2),
    # ('code', 1),
    # ('word', 1),
    # ('appear', 1)]
```

但是 python3 里面的 items()是达不到这个效果的，所以我就自己写吧。

读取 txt 文件，将每一行都按照空格划分成一个个数组的元素：

```
arr = []
   file = open('../tagcloud/tag_file.txt', 'r')
   data = file.read().split('\r\n')
   for content in data:
       contents = validatecontent(content).split()
       for word in contents:
           arr.append(word)
```

```
['BAISC', 'Python', 'BASICA', 'GVBASIC', 'GWBASIC', 'Python', 'ETBASIC', 'QBASIC', 'Quick',
'Basic', 'Turbo', 'Basic', 'True', 'Python', 'java', 'Basic', 'Visual', 'Basic', 'Visual',
'Basic', 'Net', 'Power', 'Basic', 'Python', 'java', 'SQL', 'VB', 'Small', 'Basic', 'Free',
'Basic', 'DarkBASIC', 'VBScript', 'Visual', 'Basic', 'For', 'ApplicationsVBA', 'REALbasic',
'C', 'C', 'Turbo', 'C', 'Python', 'java', 'SQL', 'VB', 'PHP', 'HTML', 'Borland', 'C', 'C',
'Builder', 'CCLI', 'Python', 'java', 'ObjectiveC', 'C#', 'Microsoft', 'Visual', 'C', 'Pascal',
'Delphi', 'Turbo', 'Python', 'java', 'SQL', 'VB', 'PHP', 'HTML', 'Pascal', 'Object', 'Pascal',
'Free', 'Pascal', 'Lazarus', 'FORTRAN', 'MATLAB', 'Scilab', 'GNU', 'Octave', 'R', 'SPlus',
'Mathematica', 'Maple', 'Python', 'java', 'SQL', 'VB', 'PHP', 'HTML', 'Julia', 'xBaseClipper',
'Visual', 'FoxPro', 'SQLPLSQL', 'TSQL', 'SQLPSM', 'LINQ', 'Xquer', 'Lua', 'Python', 'java',
'SQL', 'VB', 'Perl', 'PHP', 'Python', 'Ruby', 'ASP', 'JSP', 'TclTk', 'VBScript', 'AppleScript',
'AAuto', 'ActionScript', 'DMDScript', 'ECMAScript', 'JavaScript', 'JScript', 'TypeScript',
'sh', 'bash', 'Python', 'java', 'SQL', 'VB', 'PHP', 'HTML', 'sed', 'awk', 'PowerShell', 'csh',
'tcsh', 'ksh', 'zsh', 'XMLSVG', 'XML', 'Schema', 'Python', 'java', 'XSLT', 'XHTML', 'MathML',
'XAML', 'SSML', 'SGML', 'HTML', 'Python', 'java', 'SQL', 'VB', 'Curl', 'SVG', 'XML', 'Schema',
'XSLT', 'XHTML', 'MathML', 'XAML', 'SSML', 'Java', 'Jython', 'JRuby', 'JScheme', 'Groovy',
'Kawa', 'Scala', 'Clojure', 'ALGOL', 'APLJ', 'Ada', 'Falcon', 'Forth', 'Io', 'MUMPS', 'PLI',
'PostScript', 'REXX', 'SAC', 'Self', 'Simula', 'Swift', 'IronPython', 'IronRuby', 'COBOL',
'Python', 'java', 'SQL', 'VB', 'PHP', 'HTML']
```

对每一个元素都来个计数：

```
from collections import Counter
counts = Counter(arr).items()
print(counts)
```

效果出来了：

```
dict_items([('For', 1), ('SQL', 8), ('JRuby', 1), ('Builder', 1), ('HTML', 6), ('LINQ', 1),
('BAISC', 1), ('BASICA', 1), ('PHP', 6), ('Octave', 1), ('csh', 1), ('PostScript', 1), ('awk',
1), ('Ruby', 1), ('AppleScript', 1), ('Object', 1), ('java', 11), ('TclTk', 1), ('Xquer', 1),
('ksh', 1), ('zsh', 1), ('ETBASIC', 1), ('AAuto', 1), ('Borland', 1), ('SVG', 1), ('Jython',
1), ('Simula', 1), ('IronPython', 1), ('Python', 14), ('Microsoft', 1), ('ActionScript', 1),
('XHTML', 2), ('REXX', 1), ('COBOL', 1), ('Scilab', 1), ('Ada', 1), ('Basic', 9), ('GVBASIC',
1), ('ECMAScript', 1), ('TypeScript', 1), ('Falcon', 1), ('Clojure', 1), ('ASP', 1), ('ALGOL',
1), ('XMLSVG', 1), ('GWBASIC', 1), ('VBScript', 2), ('CCLI', 1), ('Lazarus', 1), ('Julia', 1),
('JSP', 1), ('PowerShell', 1), ('IronRuby', 1), ('Power', 1), ('FORTRAN', 1), ('Self', 1),
('Perl', 1), ('Small', 1), ('FoxPro', 1), ('REALbasic', 1), ('GNU', 1), ('Mathematica', 1),
('True', 1), ('Visual', 5), ('JScheme', 1), ('Maple', 1), ('Quick', 1), ('Turbo', 3), ('SAC',
1), ('JScript', 1), ('APLJ', 1), ('sh', 1), ('Kawa', 1), ('Pascal', 4), ('TSQL', 1), ('SPlus',
1), ('C', 6), ('xBaseClipper', 1), ('tcsh', 1), ('SQLPSM', 1), ('ApplicationsVBA', 1), ('SSML',
2), ('R', 1), ('Groovy', 1), ('XSLT', 2), ('MUMPS', 1), ('bash', 1), ('DarkBASIC', 1), ('SGML',
1), ('XAML', 2), ('VB', 8), ('Curl', 1), ('Schema', 2), ('MATLAB', 1), ('MathML', 2), ('Lua',
1), ('Net', 1), ('ObjectiveC', 1), ('JavaScript', 1), ('Java', 1), ('Io', 1), ('Free', 2),
('Delphi', 1), ('sed', 1), ('XML', 2), ('Forth', 1), ('C#', 1), ('SQLPLSQL', 1), ('QBASIC',
1), ('DMDScript', 1), ('Swift', 1), ('Scala', 1), ('PLI', 1)])
```

最后直接代入进去就行了：

```
tags = make_tags(counts, maxsize=120)

create_tag_image(tags, 'cloud_large.png', size=(900, 600), fontname='Lobster')
```

具体的修正需要自己慢慢去琢磨了，比如文字大小、图片大小、背景颜色等等。

到这里标签云是算完成了的，但是却是不支持中文，原因是没有合适的 ttf 字体文件，准备一个 ttf 中文字体，如 simhei.ttf 或者 MicrosoftYaHei.ttf ，将其移动到

```
# C:\Python34\Lib\site-packages\pytagcloud\fonts
```

接着就是更改 fonts.json 文件，按照样式添加类似于 css 的东西：

```
{

       "name": "MicrosoftYaHei",

       "ttf": "MicrosoftYaHei.ttf",

       "web": "none"

   }
```

注意前后的逗号就行。最后将这里的代码改一下：

```
create_tag_image(tags, 'cloud_large.png', size=(900, 600), fontname='MicrosoftYaHei')
```

运行，搞定！中文效果图：

我的在 github 里面，可以去下载看看。