

实用python编程 第9讲

互联网应用

2017-12-11

内容

- 搭建一个图像搜索引擎
 - 基于标签搜索图像
 - `html`介绍
 - `web.py`搭建网站

准备数据

- 下载 `flickr10k` 数据并解压缩
 - 图像（一万个jpg文件）
 - 标签（一个文本文件）

一个图像搜索引擎

- 用标签搜图像



回顾：课后练习1--统计词频

- 标签文件每一行记录了一张图片所对应的用户标签

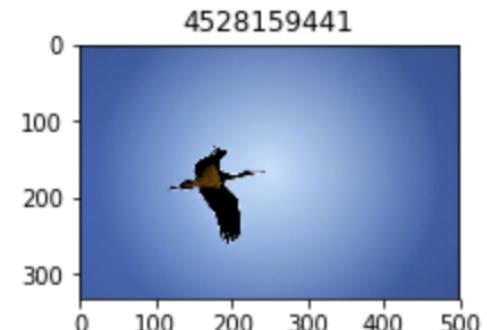
```
cedar tree grecia greece sunset sombra shadow  
black stork  
fall autumn olympus e1 nature landscape 14-54 favorites fave interesting  
organic earth getty photo scotteverett ishpop  
colbert flag gloves cascade 220 wool christmas red atlantic white  
...
```

完整数据

- 每一行第1列图像id， 第2列用户id， 第3列标签，以tab为分割符

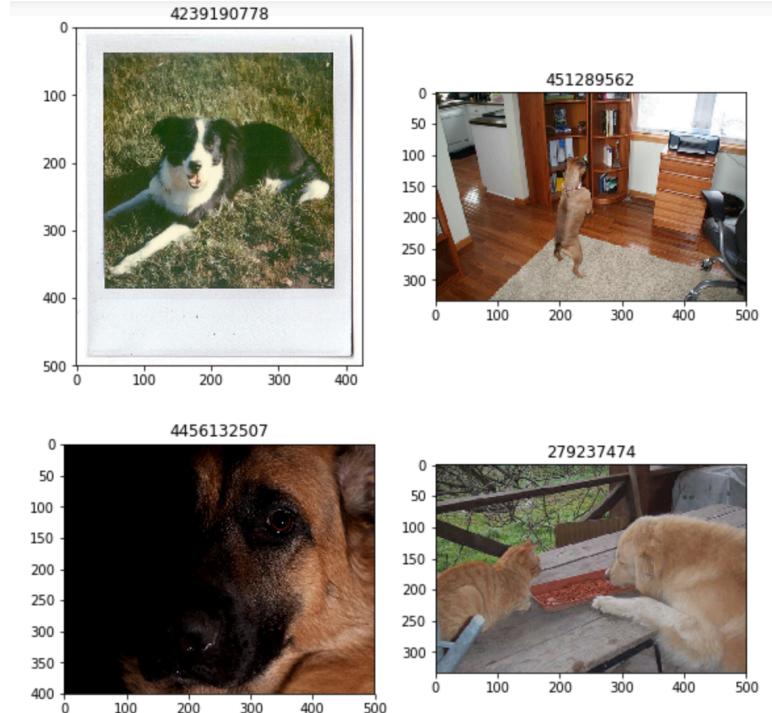
```
3949407308 27129669@N07 cedar tree grecia greece sunset sombra shadow
4528159441 63781227@N00 black stork
386023202 48604104@N00 fall autumn olympus e1 nature landscape 14-54 favorites fave
2220109467 7362777@N02 colbert flag gloves cascade 220 wool christmas red atlantic white
```

```
imageid = '4528159441'
img = mpimg.imread(get_img_path(imageid))
fig, ax1 = plt.subplots(figsize=(3,3))
imgplot = ax1.imshow(img)
ax1.set_title(imageid)
```



课堂练习：基于标签搜索图像

- 给定一个查询标签（query tag），找到所有带有该标签的图像id
 - query = 'dog' （共有209张图像）



用类 (class) 包装

- 定义一个ImageSearch类，它的search方法实现我们想要的功能

```
class ImageSearch:  
    def __init__(self, tagfile):  
        self.data = map(str.strip, open(tagfile).readlines())  
  
    def search(self, query):  
        s_time = time.time()  
        hitlist = []  
        for line in self.data:  
            imageid, userid, tags = line.strip().split('\t') # tab character as separator  
            tagset = set(tags.split())  
            if query in tagset:  
                hitlist.append(imageid)  
        timecost = time.time() - s_time  
        return {'content':hitlist, 'time':timecost}  
  
searcher = ImageSearch(tagfile)  
results = searcher.search('dog')  
print '%d hits, %.4f seconds' % (len(results['content']), results['time'])  
  
209 hits, 0.0349 seconds
```

加速图像搜索

- 倒排表
 - 以更多的内存开销换取速度的提升

dog → im1, im2, …, imN

cat → im5, im7

…

用我们学过的哪种数据类型存储倒排表？

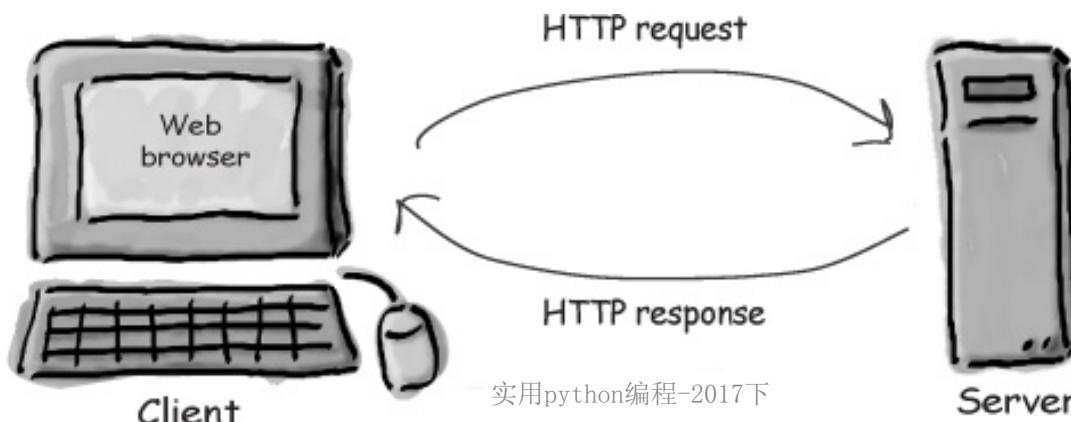
内 容

- 搭建一个图像搜索引擎
 - 基于标签搜索图像
 - [html介绍](#)
 - [web.py搭建网站](#)

在网页中展示图像搜索结果

- 首先用python实现一个简单的图像服务器
 - 从github课程网站下载代码 `code/image_server.py`
 - 根据你的flickr10k路径修改`imagedata_dir`变量
 - 启动服务器（端口9000） `python image_server.py 9000`
 - 打开浏览器，在地址栏中输入

`http://localhost:9000/img/3949407308.jpg`



网页是用html语言写的

- 超文本标记语言（英语：HyperText Markup Language，简称：`html`）是一种用于创建网页的标准标记语言

```
<html>
<body>
hello world
</body>
</html>
```

存为 `hello.html`
并用浏览器看打开



html标签

- html标签以尖括号括起来，开始标签与结束标签成对出现
 - 开始标签 <html> <body> <p>
 - 结束标签 </html> </body> </p>
 - 嵌套：一对标签可以被包含在另一对标签中
 - 特殊标签，只能出现一次：html, head, title, body

```
<html>
<body>
hello <font color="red">world</font>
</body>
</html>
```

我们需要的html标签

- 图像<**img**>标签
 - http://www.w3school.com.cn/tags/tag_img.asp
- 表格<**table**>标签
 - http://www.w3school.com.cn/tags/tag_table.asp

在html中显示多张图像

- 表格

```
<table>
  <tr>
    <td></img></td>
    <td></img></td>
    <td></img></td>
    <td></img></td>
  </tr>
  <tr>
    <td></img></td>
    <td></img></td>
    <td></img></td>
    <td></img></td>
  </tr>
</table>
```

课堂练习：将图像搜索结果写到html文件

- python image_search.py → dog.html

内容

- 搭建一个图像搜索引擎
 - 基于标签搜索图像
 - `html`介绍
 - `web.py`搭建网站

动态产生网页

- web.py框架 <http://webpy.org/>

```
pip install web.py
```

```
import web

urls = (
    '(.*)', 'hello'
)
app = web.application(urls, globals())

class hello:
    def GET(self, name):
        if not name:
            name = 'World'
        return 'Hello, ' + name + '!'

if __name__ == "__main__":
    app.run()
```

示例

- hello_world.py
- hello_bob.py
- hello_x.py
- show_images.py
- image_search.py
- image_search_form.py