

# 实验六-七报告

## 1.web.py

### 1.1. 原理简述

Python的迷人之处在于丰富的库，可以帮助处理各方面的任务。比如做网站，许多著名网站都是用Python开发的，比如YouTube、Quora、知乎、网易、豆瓣等。用Python开发网站，可以使用重量级的Zope、Django等，而对于个人的轻量使用，这次使用的web.py则是一个非常合适的Python web框架，它简单而且功能强大。框架，即framework。其实就是某种应用的半成品，把不同应用程序中有共性的一些东西抽取出来，做成一个半成品程序，这样的半成品就是所谓的程序框架。

任何网站最重要的部分就是它的URL结构。而在webpy框架中，我们就可以基于URL在python中写网站。import web之后，我们需要把我们的URL结构告诉web.py：

```
urls = (
    '/', 'index'
)
```

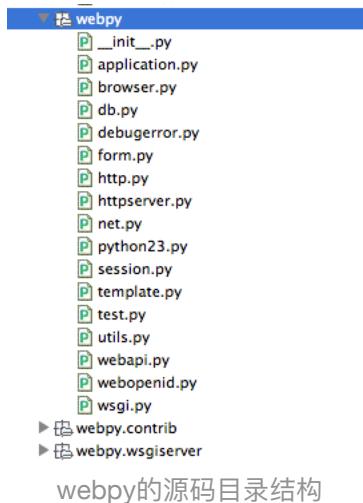
第一部分是匹配URL的正则表达式，第二部分则是接受请求的类名称。这表示我们要URL/(首页)被一个叫index的类处理。而index类部分，我们就可以自由发挥。其后我们创建一个列举这些url的application。

```
app = web.application(urls, globals())
```

也就是告诉web.py去创建一个基于我们刚提交的URL列表的application。这个application会在这个文件的全局命名空间中查找对应类。而最终我们使用templates模板进行数据交换，将网页呈现出来。

### 1.2. 源码尝试

闲来无事，尝试着阅读了webpy的源码，只有千来行，确实比较轻量。



从helloworld程序入手，找到后续我们每个main函数部分都会使用的application.run()，也就是程序的启动入口：

```
def run(self, *middleware):
    return wsgi.runwsgi(self.wsgifunc(*middleware))
```

然后跳转到runwsgi部分的runwsgi(), 也就是将各种传入的参数进行相应的初始化，并且一同启动部分的application(func)传入httpservice.runsimple()函数：

```
def runsimple(func, server_address=("0.0.0.0", 8080)):#默认的端口，若被占用  
也可自己更改  
    global server  
    func = StaticMiddleware(func)  
    func = LogMiddleware(func)  
  
    server = WSGIServer(server_address, func)  
  
    if server.ssl_adapter:  
        print "https://%s:%d/" % server_address  
    else:  
        print "http://%s:%d/" % server_address  
  
    try:  
        server.start()  
    except (KeyboardInterrupt, SystemExit):  
        server.stop()  
        server = None
```

这就是为何运行程序后有内置的web服务器。而分析StaticMiddleware类，是用来设置并加入静态路由（储存在“/static/”）的，而LogMiddleware类则是加入动态路由。同时，application本身也可以理解为一个路由，也就是后续会使用的子应用，application内部还可以添加一层application。而wsgi部分的代码是来自Python中的wsgi协议，也就是Web Server Gateway Interface，主要包括server和application两部分：server负责从客户端接收请求，将request转发给application，将application返回的response返回给客户端；而application接收由server转发的request，处理请求，并将处理结果返回给server。更深入的web开发部分我就没有继续研究了。读一读源码确实能有助于条分缕析库的内部功能实现和传递流程。

### 1.3.Aaron Swartz

值得一提的是，webpy的作者Aaron Swartz是一个年少成名的计算机天才，RSS规范的创建者，Reddit联合创始人。作为一个自由主义者，他致力于网络信息开放，在JSTOR非法下载并公开论文到学术界，却被起诉，最终在26岁的年龄用自杀的方式宣告自己的立场：用生命捍卫互联网的开放和自由。使用webpy的时候我们理应铭记Aaron Swartz。



Aaron Swartz

## 2.简单的搜索引擎

### 2.1.实现思路

在之前的几次作业中，我们已经爬取了大量文字网页和图片网页，并且对网页元素进行了分词、索引，可以使用Lucene建立简单的搜索程序了。这可以视作后端的结构完成。而在这次实验中，我们要使用webpy建立前端网页，将整个搜索、展示结果的过程在浏览器中可视化完成。

搜索和展示结果是两个不同的界面，相应地，我也建立了两个不同的类。以文本搜索为例：

```
class search_text:  
    def GET(self):  
        f = login()  
        return render.search_text(f)
```

这是文本搜索页面。其中login()对象是web.form库中的输入框，将f作为参数传递给templates中的search\_text.html文件，而search\_text.html文件中最关键的部分是：

```
<form action="/s" method="GET">  
<input type="text" id='txt' name="keyword" placeholder="关键词" >  
<button type="submit" name="search" value="搜索"></button>
```

也就是将输入的内容以'GET'方式传到'/s'页，'/s'页也就是定义的'text'类，展示文本搜索结果：

```
class text:  
    def GET(self):  
        user_data = web.input()  
        keyword=user_data.keyword  
        finalDocs = SearchTextCommand(keyword)  
        f=login()  
        return render.text(f,keyword,finalDocs)
```

其中，`web.input()`我的理解就是URL该类后面的部分，比如在text类（URL为's'）内分析`http://0.0.0.0:8080/s?keyword=你好&search=搜索`，那么`web.input()`对象的`keyword`属性就为'你好'，`search`属性就为'搜索'。值得注意的是，`keyword`部分处理空格（分开不同的搜索词）时方式应该与后面的处理方式相同，比如百度搜索就是替换成加号。如此，拿到`keyword`后，`SearchTextCommand(keyword)`函数会对`keyword`进行分词、搜索，并将得到的每个结果写成一个字典，储存了该结果的标题、关键词上下文、超链接、匹配得分等，而这些字典放在一个list中被返回。将这个大list连同着关键词、新的搜索框传入到text模板，而在text模板中，开头声明`$def with (form,keyword,finalDocs)`，就可以使用这些传入参数的各项属性并将之显示在页面上了。至此初步框架已经打成。

### 2.2.难点细节

#### ①防止多次搜索线程崩溃

我们需要把`vm_env = initVM()`一句写在主函数中，每次搜索时使用`vm_env.attachCurrentThread()`新建线程，以阻止多次搜索程序崩溃。相比直接写在主函数中，我们发现更好的办法是将`initVM()`部分写在另一个Python文件`initvm.py`中，在搜索程序中直接`import initvm`调用之，其后每次要进行搜索时，`initvm.vm_env.attachCurrentThread()`

可以确保是新建线程。并且这样写还有一个优势，切换图片或文字搜索时可以方便地换用储存index的路径，进而更新searcher和analyzer。

### ②直接传入html语法数据

前文中已经写过，执行函数SearchTextCommand(keyword)返回的是一个大list，储存了每个搜索结果的dictionary，而这些结果的超链接、上下文高亮等，我直接写成了html格式的数据，也就是在Python中写的并传入，比如

```
contentStr+= '<font color="red">'  
...  
contentStr+= '</font>'
```

然后将contentStr整个字符串作为该结果的内容。此时在模板展示中，若仍然使用\$oneDic['content']，其中会与HTML有关的语法符号会被替换（防止二义性）。若要不被替换，需要写成\$:\$oneDic['content']，加冒号表示直接引用，不转义。

### ③关键词高亮+寻找上下文

我的初步想法是找到结果的path之后，打开储存的对应的页面，使用bs分析得到页面中的文字部分，然后对于每一个搜索词，使用正则搜到出现该词的上下文20个字，然后对于这个字符串进行②中所示处理，最后对于得到的优先的10个以内字符串作为该结果的content储存在dictionary中，并传递给展示页面。

```
path=doc.get('path')#html文件夹太大未上传，故文本搜索在助教电脑上无法执行  
file = open(path)  
contents = file.read()  
soup = bs4.BeautifulSoup(contents,features="html.parser")  
contents = ''.join(soup.findAll(text=True))  
file.close()  
allcontentlist=[ ]  
  
for oneContent in contentCommand:  
    wordlen=len(oneContent)  
    contentlist=[i.start() for i in re.finditer(oneContent,  
contents)]  
    for location in contentlist:  
        if(location>20):  
            allcontentlist.append((location,wordlen))  
    allcontentlist.sort(key= lambda k:k[0])  
  
    maxlen=min(len(allcontentlist),10)  
    contentStr=""  
    for i in range(maxlen):  
        location=allcontentlist[i]  
        for i in range(location[0]-10,location[0]):  
            if (contents[i]!=' ' and contents[i]!='\n' and  
contents[i]!='\t'):  
                contentStr+=contents[i]  
    contentStr+.'<font color="red">'  
    for i in range(location[0],location[0]+location[1]):  
        contentStr+=contents[i]  
    contentStr+.'</font>'
```

```

        for i in
range(location[0]+location[1],location[0]+location[1]+10):
    if (contents[i]!=' ' and contents[i]!='\n' and
contents[i]!='\t'):
        contentStr+=contents[i]
oneDic['content']=contentStr

```

刚开始我觉得可能很耗时，但实际写出来后，一般的搜索文本都能在1s之内处理完，可以接受。或许是因为目前的搜索还比较轻量。但在工业界应用这样显然是愚蠢的，不仅要本地储存耗费空间，而且大型运行会耗时大。因此考虑优化。

优化的方向可以在索引关键词的时候，对每个关键词的上下文（也就是其他的关键词）同时进行相对位置的索引，然后结合MySQL使用。后来和同学讨论发现，可以利用Lucene自带的库和函数来获得关键词的上下文。前提是要把相关的内容索引储存，也就是建立倒排的原理。

### ③处理英文字符

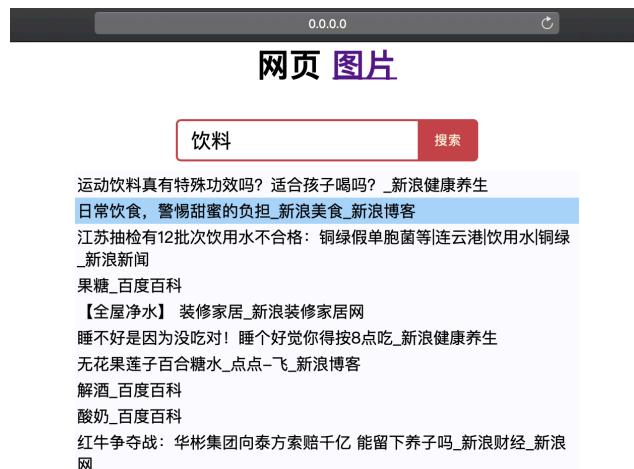
为了获得网页的文本内容，我一直使用的是contents =

'''.join(soup.findAll(text=True))语句。这样的确可以去除掉所有html标签，但是对于内嵌的js代码等部分，仍然无法消除，这样造成文本储存的索引仍然有很多英文字符存在。但是我也不能直接暴力地只采用网页中的中文部分，去除所有英文字符，因为网页也可能是本身是英文内容。这个问题仍然存在，还在考虑怎样优化。

## 3. 使用div+css规范网页样式

CSS 指层叠样式表 (Cascading Style Sheets)，是一种用来表现HTML（标准通用标记语言的一个应用）或XML（标准通用标记语言的一个子集）等文件样式的计算机语言；而DIV+CSS为“WEB标准”中常用术语之一。DIV是用于搭建html网页结构（框架）标签，像<b>、<h1>、<span>等html标签一样，CSS用于创建网页表现（样式/美化），也就是通过css来设置div标签样式。可以参见CSS属性手册，使用CSS规定多种样式，包括高度、宽度、边框、背景等等。

坦率地说，我一直不太喜欢也不太擅长做前端设计。虽然可以像画画一样自己设计网页，并且不用debug，直接可视，但写前端的过程总是实际和所想有较大差距，几次下来就有些抵触。这次要感谢我的室友杨子超同学，是在他的帮助和讲解下我顺利完成了前端的设计。展示图如下：



文本搜索框和自动提示框，鼠标悬浮某条提示上面可以变蓝并点进

**牛奶 (饮品) - 百度百科**

OK 牛奶 (饮品) - 百度百科 牛奶 是一个多义词, 请在下拉框中选择。 牛奶 顾名思义是从雌性奶牛出来的。在不同国家, 牛奶 也分有不同的等级。牛奶也分有不同的等级。 牛奶 含有丰富的矿物质、钙、镁、钼。最难得的是, 牛奶 是人体钙的最佳来源, 等。

牛奶 可以助眠么? 睡人推荐你睡前喝一杯热牛奶, 有人说效果不错, 也

<https://baike.baidu.com/item/%E7%89%9B%E5%A5%B6/28828>

**水牛奶 - 百度百科**

OK 水牛奶 - 百度百科 水牛奶 编辑锁定编辑锁定水 牛奶 , 与人们熟知的乳用牛奶 (牛) 产的奶不同。水 牛奶 产量较低, 营养价值高研部门测定, 1 公斤水 牛奶 所含营养价值相当于黑营养价值相当于黑白花 牛奶 1.83 公斤, 最适宜、双皮奶, 都必须用水 牛奶 来制作。 [日] 中文名水 牛奶 外文名Wat水 牛奶 营养价值编辑水营养价值编辑水 牛奶 的脂肪、蛋白质、乳糖

<https://baike.baidu.com/item/%E6%B0%B4%E7%89%9B%E5%A5%B6>

**双皮奶 - 百度百科**

咕噜视频: 牛奶 的三种吃法朝时起源自顺德, 用牛奶 做原料; 现遍布于广东碗上佳的双皮奶, 以水 牛奶 做原料, 其状如膏, 其主要食材 牛奶 , 鸡蛋清, 白糖; 餐的时候, 不小心在水 牛奶 做了个花样, 无意中做法-食材鲜 牛奶 500ml, 鸡蛋3个糖随意。步骤1. 牛奶 倒入锅中加热, 但不要奶2. 加热后的鲜 牛奶 倒入容器中, 放在通风成(这一层奶皮是靠鲜 牛奶 的热气往上顶, 而形成。4. 把奶皮下面的 牛奶 倒出, 留少许牛奶在碗

[https://baike.baidu.com/view/33935.htm#ref\\_\[3\]\\_33935](https://baike.baidu.com/view/33935.htm#ref_[3]_33935)

**酸奶 - 百度百科**

锁定酸奶是以 牛奶 为原料, 经过巴氏杀菌, 经过巴氏杀菌后再向 牛奶 中添加有益菌(发酵剂后, 再冷却灌装的一种 牛奶 制品。目前市场上酸奶为多。酸奶不但保留了 牛奶 的所有优点, 而且某些未必安全酸奶是 牛奶 经过乳酸菌发酵而成的所以发酵前必须对原料 牛奶 和发酵器具进行杀菌主要原料鲜 牛奶 , 白糖, 全脂奶粉, 水或瓷瓶或专用纸盒、鲜 牛奶 、白糖、乳酸菌种(保器内灭菌30分钟。 牛奶 灭菌。把鲜牛奶装入加钟。牛奶灭菌。把鲜 牛奶 装入加热罐, 并加入1

<https://baike.baidu.com/subview/11230/4925129.htm>

**奶酪 (奶制品) - 百度百科**

名干酪, 是一种发酵的 牛奶 制品, 其性质与常见的奶, 其性质与常见的酸 牛奶 有相似之处, 都是通过制品都是由10公斤的 牛奶 浓缩而成, 含有丰富的酸而言, 奶酪是发酵的 牛奶 ; 就营养而言, 奶酪是养而言, 奶酪是浓缩的 牛奶 。奶酪也是中国西北的使用63℃低温杀菌的 牛奶 , 因为经过高温消毒的, 因为经过高温消毒的 牛奶 已经不能作为原料。如牛很多水, 那样会降低 牛奶 质量。另外一定要劝诫供应商, 不要试图在 牛奶 中添加牛奶的副产物, 高温发酵菌种, 在加热 牛奶 的过程中, 低于发酵温

<https://baike.baidu.com/item/%E4%99%B3%E9%85%AA>

**奶皮 - 百度百科**

奶皮的制法是将鲜 牛奶 入锅煮熟后, 控制火力两种: 一种是将鲜 牛奶 或马奶、驼奶、羊奶放锅, 是农牧地区农民用 牛奶 制成的著名土特产品之著名土特产品之一, 为 牛奶 制品中最可口的营养奶皮原料: 全脂 牛奶 , 鸡蛋2只(小型), 打单散备用。3. 将 牛奶 倒入小锅中, 用小火加。4. 将煮得略滚的 牛奶 从锅中倒入碗中。5. 牛奶 在室温中自然冷却约2了。8. 留一丁点 牛奶 在碗中, 这样奶皮轻轻事先打散的蛋白液倒入 牛奶 中。10, 加入适量

<https://baike.baidu.com/item/%E5%A5%B6%E7%9A%AE>

**【可颂&可颂三明治】28℃的可颂 - 百度的小厨房 - 新浪博客**

砂糖20克, 盐5克, 牛奶25, 水120克, 黄 【 牛奶 大方包】 【 北海道 牛 奶士司】 【 牛奶 排包】 香杏仁土司】 【 牛奶 吐司 & 香蒜吐司条】 【 牛奶 豆沙排包】 【 牛奶 天使面包】 【 巧克力 & 牛奶 坚果小方包】 【 果粒 牛奶 面包和海苔肉松包】 【 牛奶 豆沙面包排】

[http://blog.sina.com.cn/s/blog\\_8738c14f0102xty.html](http://blog.sina.com.cn/s/blog_8738c14f0102xty.html)

**酥酪 - 百度百科**

奶制品, 主要用羊奶、 牛奶 等制成, 故又有“乳酪”料羊奶、 牛奶 味道足的最高境界。乳 牛奶 酪, 牛奶凝乳, 即乳, 牛奶, 酪, 牛奶 凝乳, 即凝结的牛奶, 牛奶凝乳, 即结的 牛奶 , 生酥, 新鲜奶酪, 了。且别说我吃了一碗 牛奶 , 就是再比这值钱的, 该的。”糖蒸酥酪即指 牛奶 无疑。牛奶在当时是糖蒸酥酪即指 牛奶 无疑。 牛奶 在当时是稀贵之物, 并, 《燕都小仪器杂咏· 牛奶 酪》云: “鲜新美味矣, 味颇美, 制此者为 牛奶 房也。酥

<https://baike.baidu.com/item/%E9%85%A5%E9%85%AA>

**奶山羊 - 百度百科**

干物质11.6%。与 牛奶 相比, 羊奶含干物质、生素C、尼克酸均高于 牛奶 , 不仅营养丰富, 而且人奶相似, 酸值低, 比 牛奶 易为人体吸收。是要干物质11.6%。与 牛奶 相比, 羊奶含干物质、生素C、尼克酸均高于 牛奶 , 不仅营养丰富, 而且人奶相似, 酸值低, 比 牛奶 易为人体吸收。是要

<https://baike.baidu.com/item/%E5%A5%B6%E5%85%81%BE%8A>

## 网页 图片

牛奶 搜索

【欧乳菲牛奶】德国进口牛奶 欧乳菲（EURO FIT）全脂纯牛奶 1L\*12盒  
【八享时桃酥牛奶味100份】桃酥牛奶味100份  
【香满楼牛奶】香满楼 纯牛奶 250ml\*16 (新老包装随机发货)  
【光明牛奶】光明 优+纯牛奶250ml\*12盒/礼盒装中华老字号  
【佳乐锭牛奶片】意大利进口 Galatine佳乐锭/阿拉丁牛奶片 100g  
【蒙牛袋装红枣酸牛奶150g\*15袋】蒙牛 (MENGNIU) 风味发酵乳 红枣风味酸奶 150g\*15整箱装  
【土耳其进口 优客阿尔巴尼-牛奶心巧克力 60g】土耳其进口 优客 (Ulker) 阿尔巴尼-牛奶夹心巧克力 60g  
【巧乐牛巧乐牛香蕉牛奶（调制乳）礼盒250ml\*6瓶 礼盒】奥地利进口 巧乐牛 JOLLY COW 巧乐牛香蕉牛奶（调制乳）礼盒250ml\*6瓶 礼盒  
【Lifefactory奶嘴奶瓶】Lifefactory Y字奶嘴  
【小迷糊爽肤水】小迷糊 牛奶柔滑亮肤精华水110ml (提亮肤色 细腻嫩滑)

## 图片搜索界面



图片搜索结果页面，鼠标悬浮在某图片上会变成半透明提示

## 4.附加功能：自动实时提示

### 4.1.功能简述

对比目前我的搜索功能和百度搜索，我认为很大一个差别在于百度有自动提示。这样在搜到一半的时候很可能就直接看到了想要的结果，大大提高了效率。那么如何模仿百度的自动补全呢？去年电工导我们使用了Ajax的方法，但是用过了就没有新意了。而且，使用Chrome浏览器抓包并分析发现，百度使用的并不是调用Ajax接口，而是用的js方法。正巧，上次我爬取动态网页也对js语句有了初步的了解，这次使用js可以加深了解。

ps.后来在网上看了，为何百度谷歌搜索的自动补全不使用Ajax呢，因为许多其他站点比如导航网站，也要使用百度搜索的接口，如果是用Ajax去调用的，根据同源策略就取不到数据。（也有地方说除非设置了CORS）。

## 4.2. 实现步骤

首先百度的实时提示使用的是js传回包，我用Chrome浏览器打开页面并监视network部分，当我在输入框输入字符时，实时就会传回数据。比如当我输入上海，页面显示如下：



而监视返回的包：

The screenshot shows the Network tab of the Chrome DevTools. A single request is highlighted: "Request URL: https://www.baidu.com/su?&wd=上海&p=3&cb=BaiduSuggestion.callbacks.give". The status is "200 OK". Headers show "Cache-Control: private" and "Content-Encoding: gzip". The response body is a large JSON object. At the bottom, it says "3 requests | 1.1 KB transferred".

其中这个http请求：'https://www.baidu.com/su?&wd=上海&p=3&cb=BaiduSuggestion.callbacks'，用浏览器打开，是这样一条数据（包裹json数据的函数执行）：

BaiduSuggestion.callbacks({q:"上海",p:false,s:["上海旅游","上海声明","上海天气","上海进口博览会","上海社保","上海税务","上海地图","上海地铁"]});

简单分析并替换URL中相关部件后，可知是jsonp获取数据的格式，wd是当前的关键词，传入站点'su'应该就是百度的自动补全函数，而cb后面则是回调函数的名字，将获取的数据在当前页面进行展示。这一套流程正是使用jQuery的jsonp在不同服务器端口间发起跨域请求。

这样，一套完整构想就初具雏形：

首先建立td标签对象和其对应的css，也就是提示词的容器。当容器为空时隐藏。自己先设计好webpy的类，传入查询词，返回函数包裹着jsonp数据的函数执行。再使用document类，创建'script'对象s，用keyup事件，当有输入动作时，通过调用this.value.trim()获得当前的输入，然后将这个输入放在自己定义的URL格式中，将这个URL赋给s的src，再打开s，实际上就是运行接下来的函数：展示s中的每个联想到的title，并且如果悬浮点击，会进入搜索该title的页面。最后执行完fn函数后，要将动态插入的脚本s删掉。

其中webpy写类部分，`jsontext`和`jsonimg`类，传入搜索词后，会得到10个结果的标题，这些标题仿照百度的`jsonp`格式传回。要先`web.header('content-type', 'text/json')`，以向前端返回`json`数据；同时对于建好的字典，要使用`json.dumps()`压成`json`数据包。

#### 4.3. 细节问题

- ①由于样本量限制，目前使用的是联想到搜索结果，而不是以当前开头的title。
- ②点击联想词并不是直接进入对应网页，而是对该title进行搜索。这和百度的联想搜索逻辑是一样的，但其实并不适合这里，因为已经返回专一的title了，应该直接导入对应的URL。
- ③有些title含有特殊字符，比如'【欧乳菲牛奶】德国 进口牛奶 欧乳菲 (EURO FIT) 全脂纯牛奶 1L\*12盒'，其中的''字符就会在搜索时导致错误。应该直接导入URL或者对title进行文本处理。
- ④展示图上文已有，不赘述。
- ⑤由于没有上传本地html文件，在助教电脑上可以运行联想词和完整的图片搜索，但文本搜索的结果展示无法运行。