电工导实验报告 4

F1403023　5140309534　韩坤言

一、实验目的

1. 了解 web.py 利用 web 框架进行简单的 web 开发
2. 建立一个简单的搜索引擎

二、实验内容

1. 利用 web 框架 web.py 实现 web 开发的基本功能
2. 运用模版简化工作量，运用表单达到页面跳转的功能
3. 建立一个基于之前的搜索引擎，实现网页搜索和图片搜索

三、实验环境

1. Firefox + Firebug 插件或 Chrome
2. Python 2.7 + easy_install + BeautifulSoup
3. JCC + PyLucene
4. web.py

四、实验步骤

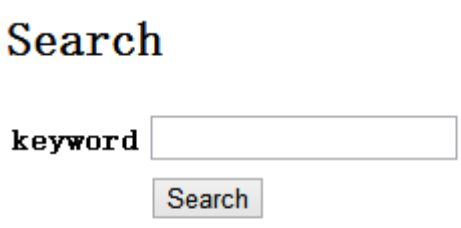
所谓 web 框架，就是某种应用的半成品，好处是减少重复开发的工作量，缩短开发时间，降低开发的成本。我们使用的是 web.py，基于 python 进行 web 开发。通过一些例子来了解各个部分是如何运作的，了解了模版，以及如何处理表单。

中期整合则是在之前几次 lab 的基础上加加以整合，为了界面的美观与整洁，运用 div+css 来规范网页的样式，以及布局，调整各个模块的属性以达到满意的效果。

五、问题及其解决

1. 使用 web.py，结合前面学习的 HTML，Lucene，中文分词等知识点，根据上次实验爬取的网页，建立一个简单的搜索引擎。

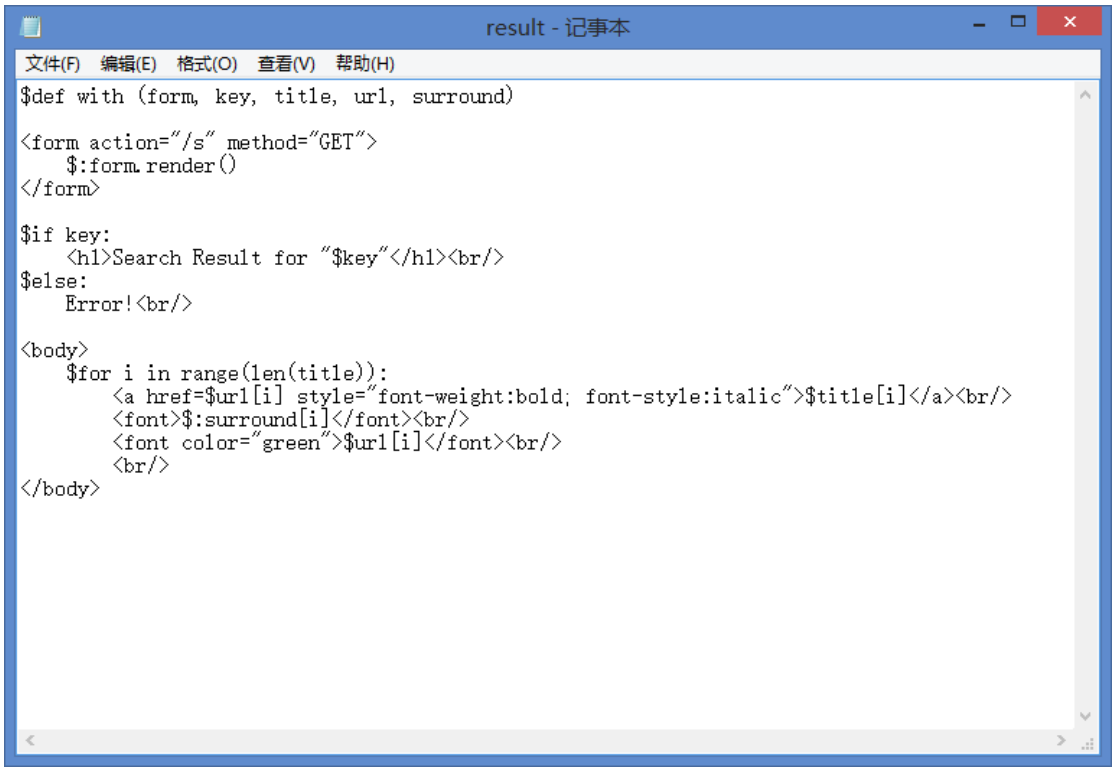
在了解了 web.py 的工作原理之后，其实这个问题也就引刃而解了。我们在做的无非就是把用户在 input 框中的 keyword 传入 Search.py 进行搜索，最后把结果通过模版的方式呈现给用户。



初始界面

结果界面

实现起来比较容易，展示搜索结果通过模版呈现，通过 for 循环，把结果依次现实。<a href=...>来创建超链接，通过改变参数来修改字体，颜色等，使其看起来像一个搜索引擎的结果页面。



result

过程中比较麻烦的就是得到关键词周围的文字信息。这势必需要把 content 保存下来，放到 index 里，考虑到这样查询的效率会很低下，索性要找周围信息的时候重新打开一次文件速度会快得多。得到前后的信息涉及到的是字符串的处理，由于用过 bs 的 get_text()后任然有许多 html 的 tag 残留，直接向前向后找是不可取的。我过滤掉了许多明显的具有 tag

标志的字符，然后先前寻找标点符号或者空格，当字数达到一定的数量以后就停止，把这一段存下来，向后同理，再把中间的关键词前后加上 html 的语言使其显示出来的时候呈红色。

```
ref = [' ','\n','"','\'',';','{','}','<','>','\r','\r\n']
while front>=0 and not other_word(content[front]):
    if content[front]!='\n' and content[front]!=' ' and content[front]!='\r' and content[front]!='\r\n':
        if content[front] not in ref:
            a = content[front] + a
        front = front - 1
    if len(a)>25 and (content[front] in ref or is_punc(content[front])):
        break;
    flag = True
    while content[front]=='\n' or content[front]==' ' or content[front]=='\r' or content[front]=='\r\n':
        if flag:
            a = " " + a
            flag = False
        front = front -1
```

```
def all_sur(command_list,content):
    a = ""
    b = ""
    surround = ""
    for item in command_list:
        a,b = find_sur(item,content)
        if a=="error" and b=="error":
            continue
        surround = a + '<font color="red">' + item + '</font>' + b + '...'
        break
    return surround
```

部分代码

虽然界面很朴素，但是基本的功能都已经实现。

2. 制作一个图片加文字的搜索引擎，作为中期整合。在上次的基础上，加入图片搜索，使用 css 制定样式

首先是把图片的搜索功能加入。我的想法是一个搜索框对应两个 button,一个搜索 website,一个搜索 picture。经过大量资料的查询,我实现了我的想法。我舍弃了之前的 form 模版，自己创建了一个 type 是 text 的输入框和两个 type 是 button 的按钮，不直接选择按钮是应为这样能添加更多的属性。通过添加鼠标的 click 事件来进行跳转。很不错的达到了预期的效果。

```
<label align="center">
    <input type="text" name="keyword" placeholder="请输入关键词" />
</label>
<span style="font-size:34px;">       </span>
<input type="button" value="网页搜索" onclick="web()" class="button" />
<input type="button" value="图片搜索" onclick="pic()" class="button" />
```

```
<script language="javaScript">
    function web(){
        form1.action="/s";
        form1.submit()
    }

    function pic(){
        form1.action="/p";
        form1.submit()
    }
</script>
```

为了美观，通过 div+css 的方式来使网页中的每个元素更加漂亮。

我定下的基调是黑色和金色的高雅风格，我选择了一张合适的背景，为了防止背景重复以及变动，我进行了如下设置

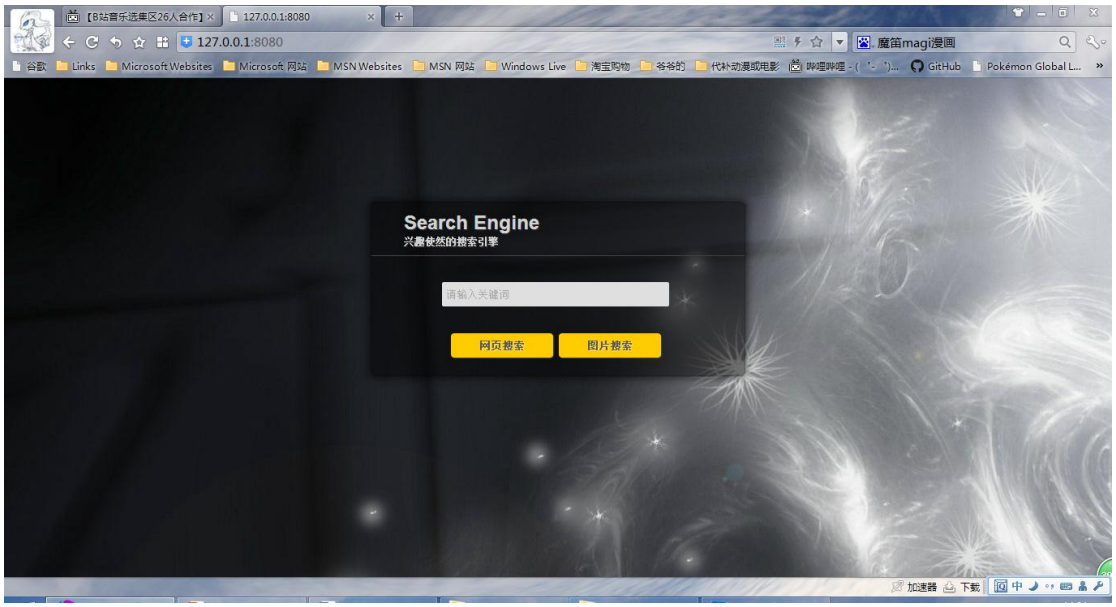
```
<style type="text/css">
    body{
        background-image: url(http://pic.pp3.cn/uploads//allimg/111125/15455S941-9.jpg)
        background-repeat: no-repeat;
        background-attachment: fixed;
    }
```

我修改了很多属性值，使得 form 的表单呈现半透明的效果，部分代码如下：

```
<head>
    <style type="text/css">
        body{
            background-image: url(http://pic.pp3.cn/uploads//allimg/111125/154
            background-repeat: no-repeat;
            background-attachment: fixed;
        }

        .dark-matter {
            margin-left: auto;
            margin-right: auto;
            max-width: 400px;
            padding: 20px 30px 20px 30px;
            font: 12px "Helvetica Neue", Helvetica, Arial, sans-serif;
            color: #D3D3D3;
            text-shadow: 1px 1px 1px #444;
            border: 1px solid rgba(0,0,0,.2);
            -moz-border-radius: 5px;
            -webkit-border-radius: 5px;
            border-radius: 5px;
            -moz-background-clip: padding;
            -webkit-background-clip: padding-box;
            background: rgba(0, 0, 0, 0.5);
            -moz-box-shadow: 0 0 13px 3px rgba(0,0,0,.5);
            -webkit-box-shadow: 0 0 13px 3px rgba(0,0,0,.5);
            box-shadow: 0 0 13px 3px rgba(0,0,0,.5);
            overflow: hidden;
        }

        .dark-matter h1 {
            padding: 0px 0px 10px 40px;
            display: block;
            border-bottom: 1px solid #444;
            margin: -10px -30px 30px -30px;
        }

        .dark-matter h1>span {
            display: block;
            font-size: 11px;
        }
```

搜索网页的结果界面比较容易，多弄几个框，放进去就行了，设置沿用之前的即可，图片搜索的结果页面为了使一排现实多个图片，我设置了 float，图片就会多张并排显示了。

最终的界面如下：

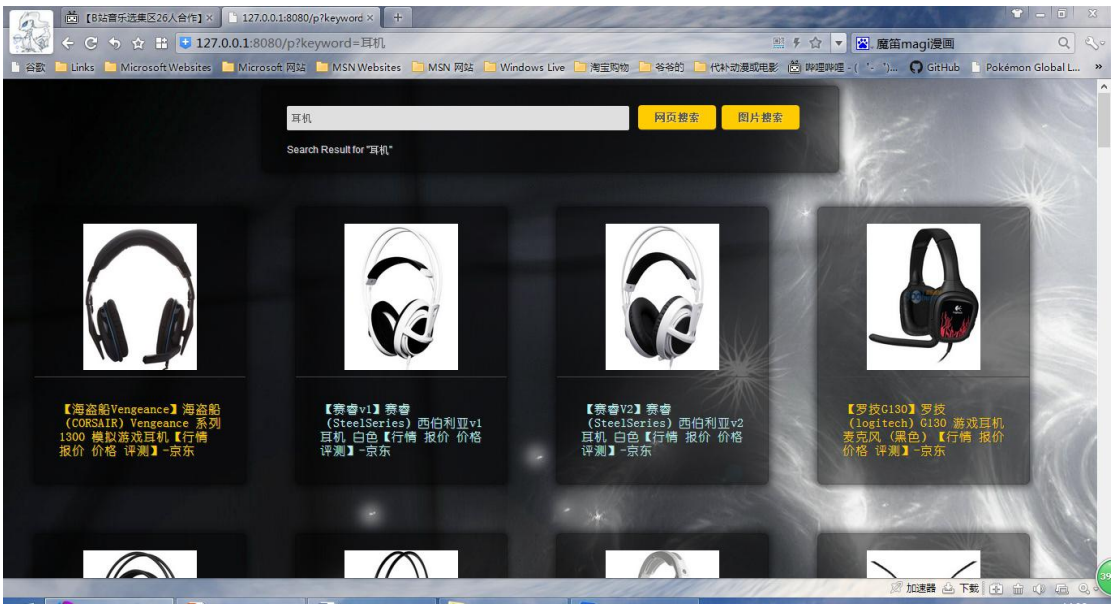


起始界面

起始界面细节展示

网页搜索结果信息界面

网页搜索结果信息界面细节展示



图片搜索结果页面

网页我爬取了 1000 个，图片我爬取了 500 个来建立索引。网页从 sjtu 开始爬，图片从京东的某游戏展区开始爬的，如果需要测试可以测试一些相关关键词。

我发现用户点击网页搜索后到搜寻结果出来有一段等候时间，这时间随我爬取的网页数增多而增加，如果有方法能提高处理的速度就更加完美了。

六、实验总结

这次的作业把之前所有的成果整合在一起，虽说遇到历史遗留问题很麻烦，但是当自己的成果展现在网页上呈现在眼前的时候，能真切的体会到自己努力后的兴奋和喜悦。其中不乏被报错信息烦的半死，也有根本不知从何处下手的情况。可到最后，一切都引刃而解的时候，成就感满满。通过这半个学期的学习和练习，我对计算机网络，网页等方面都有了极大的兴趣，学习了很多很多。我很期待后半个学期的进一步的学习。