

实验 7-中期整合

姓名：王煌基

学号: 519030910100

班级： F1903004

一、实验概览

基于对前半学期课程内容的学习（包括但不限于 HTML 语言、网络爬虫、Lucene、Flask 框架等）以及对 div+css 规范网页样式的接触、了解，尝试制作一个图片+文字的搜索引擎，并且规范其相应的格式。

二、实验环境

1. 个人笔记本电脑
2. 操作系统：windows10 专业版
3. 使用软件：Visual Studio Code；Docker Desktop

三、实验过程

练习一：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。

【解】

Search

keyword

Search Result for "闪电"

[【闪电快打小游戏大全】经典闪电快打小游戏 - 17yy小游戏](http://www.17yy.com/s/shandiankuaida)
闪电快打小游戏大全
<http://www.17yy.com/s/shandiankuaida>

[【闪电冰火人】小游戏 - 17yy经典小游戏](http://www.17yy.com/f/233840.html)
闪电冰火人小游戏经典
<http://www.17yy.com/f/233840.html>

[【闪电冰火人2中文版】小游戏 - 17yy经典小游戏](http://www.17yy.com/f/239460.html)
闪电冰火人中文版小游戏
<http://www.17yy.com/f/239460.html>

[【闪电冰火人2中文无敌版】小游戏 - 17yy经典小游戏](http://www.17yy.com/f/239461.html)
闪电冰火人中文无敌
<http://www.17yy.com/f/239461.html>

【闪电冰火人2中文终极无敌版】小游戏 - 17yy经典小游戏
闪电冰火人中文终极
<http://www.17yy.com/f/239524.html>

[【闪电快打6无敌版之黄金AK】小游戏 - 17yy经典小游戏](http://www.17yy.com/f/58003.html)
闪电快打无敌版
<http://www.17yy.com/f/58003.html>

[【闪电快打4】小游戏 - 17yy经典小游戏](http://www.17yy.com/f/57998.html)
闪电快打小游戏经典
<http://www.17yy.com/f/57998.html>



通过本次实验的参照结果以及上次实验的对比，可以得到几个不同之处，而这几个不同之处就是我们需要攻克的难题：

- ①本次网页支持不同搜索情况（图片、网页）的跳转；
- ②本次网页的显示结果均以一个类似的格式进行包装（每个内容均用一个网格包装）。

下面我将对逐个难题进行解决。

1. 本次网页支持不同搜索情况（图片、网页）的跳转：

这个难题本质上即当我们点击相应的内容时，应能够跳转到相应的页面中，如：当我们在网页搜索页面的时候，应该保证网页是黑色的，而图片字体是可点击的超链接：

```
1. <h1>Search</h1>
2. <font size = "4" >网页 <a href="/search_img" >图片</a></font>
3. <form action="result">
4. <input type="name" name="keyword"><br>
5. <input type="submit" name="搜索">
6. </form>
```

该段 HTML 的预览为（截图的原因，出现了黑线，网页中不存在黑线）：



通过限定图片的超链接跳转到 search_img 页面下即可满足我们的要求。

2. 本次网页的显示结果均以一个类似的格式进行包装（每个内容均用一个网格包装）：

由于之前并没有接触过 css，我对这方面相当的陌生。但是通过查阅相关资料后，我可以逐渐习得 css 的用法。在这个实验中，我们需要创建一个 box 来封装我们想要集成的内容，以实现参照结果中出现的每一个搜索结果都出现在一个小 box 中，因而我们需要定制一个特殊 style 的 css 代码。

根据参照结果中的对比，我们可以发现，在具体实现的过程中我们需要让我们的 box 具有几个属性的限定：1. 宽度限定，网页的搜索结果宽度基本 100%，图片的搜索结果宽度控制在 20%左右不错；2. 高度限定，由于字体的大小或者图片的高度等原因，均需要控制在合理的高度范围，防止高度过小而内容溢出；3. 颜色限定，边界颜色为黑色，底色为任意适合的颜色（这里取灰色）；4. 对齐限定，对于网页搜索，不需要对齐，因为本身就已经占满了一整行的内容，而对于图片的搜索结果，需要保证是左对齐的方式以使得一行能够出现多行的结果；5. 空隙限定，每个搜索结果的连接处均有空行的连开，因此我利用 margin-{place} 的方式来

修正我们的空隙。通过这样的方式，我们构造好了我们的 box：

```
1. .content{
2.     width: 95%;
3.     height: 120px;
4.     background-color: rgb(124, 123, 121);
5.     float:left;
6.     margin-bottom: 20px;
7.     margin-left: 10px;
8.     border:1px solid #000000;
9. }
```

并且预览结果如下（内容为 HTML 文件中已经带有的）：

```
{{ texts.title }}
{{ texts.left }}{{ keyword }}{{ texts.right }}
{{ texts.url }}
```

进而我们已经将我们本次实验中的两个难题解决了！下面是将 lab4 的网页、lab5 的图片索引的建立、搜索过程整合到我们的本次实验中。

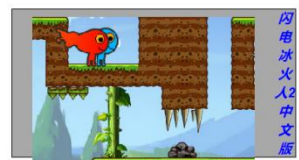
首先，由于本次实验具有“网页”、“图片”两个方向的搜索，故在建立 html 网站时，我也随之建立 4 个 html 网站(search.html、search_img.html、result.html、result_img.html)，并且利用 app 变量的 route() 装饰器来告诉 Flask 框架 URL 去触发我们的 4 个网址。

```
1. @app.route('/search', methods=['POST', 'GET'])
2. @app.route('/result', methods=['POST', 'GET'])
3. @app.route('/search_img', methods=['POST', 'GET'])
4. @app.route('/result_img', methods=['POST', 'GET'])
```

其次，建立搜索的过程是相似的，其中比较麻烦的一点在于，在 lab4、lab5 中我均利用 index 文件夹来存放我们建立好的索引，故出现了重名的情况，这里我直接将 lab5 的内容改成 index2 即可，在具体函数实现的过程中只需要特判即可，从而我们本次的实验任务解决。

四、实验问题及解决

问题一：在图片导入的过程中出现了大小不一的情况，如何调整使得能够不出现超出界面的大小？



【解】

由于刚开始对 css 语言并不熟悉，查阅了非常多资料，才知道可以通过限定图片在 box 的属性来约束图片的大小，这里仅需要建立 style 属性的 img 类，我们便可以控制图片在一个

box 内的大小:

```
1. .imgs{
2.     width:86%;
3.     height:65%;
4.     float:left;
5.     margin-top:10px;
6.     margin-left:7%;
7.     margin-right:7%;
8.     margin-bottom:10px;
9. }
```

从而, 我们的问题得到了解决:



问题二: 如何通过简单的 css 样式实现类似百度的搜索框?



【解】

通过在“菜鸟教程”中对 Bootstrap 的学习, 我们可以直接调用其中的按钮组合代码“btn-group”来实现简单的按钮组合。对其中关键代码进行修改后, 我们可以利用:

```
1. <div class="btn-group" style="float:left; margin-right: 20px;">
2.     <a class="btn btn-
   default" href="/search" role="button"><span class="glyphicon glyphicon-search"></span> 网
   页</a>
3.     <a class="btn btn-
   default" href="/search_img" role="button"><span class="glyphicon glyphicon-
   picture"></span> 图片</a>
4. </div>
```

来代替我们之前的:

```
1. <font size = "4" >网页 <a href="/search_img" >图片</a></font>
```

虽然在表面上代码更加复杂化、更加繁琐, 但是这样带来的视图效果更贴近实际生活中接触到的搜索引擎, 也更容易让人接受:

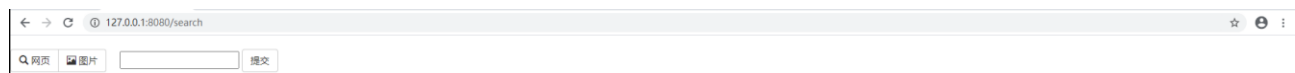


五、实验结果

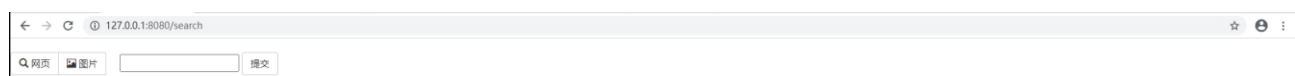
运行 search.py 建立网页：

```
^Crooot@f3e30b12bd20:/workspaces/lab7-中期整合# python search.py
* Serving Flask app "search" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 175-463-158
```

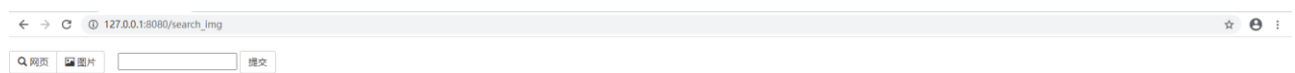
查看 127.0.0.1:8080/search:



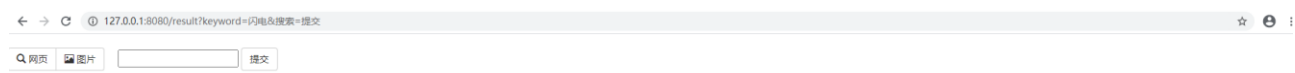
点击“网页”按钮，地址不变：



点击“图片”按钮，地址跳转到/search_img:



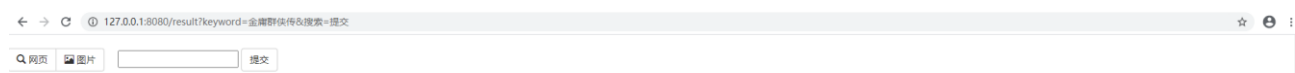
在“网页”输入为闪电：



为您找到的“闪电”的网页搜索结果有94个



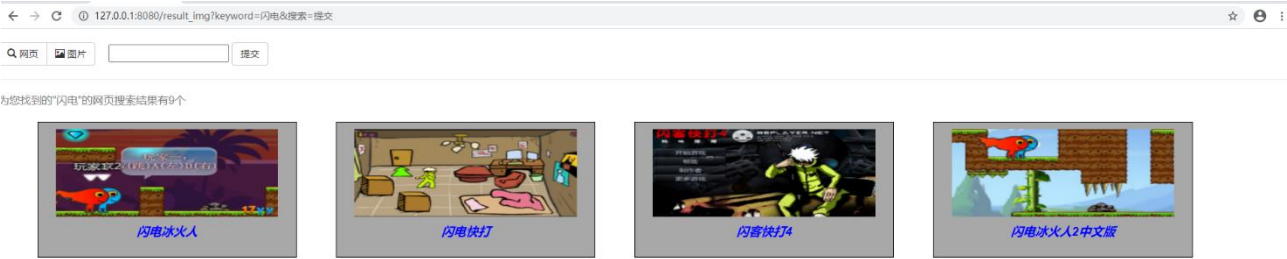
在“网页”输入为金庸群侠传：



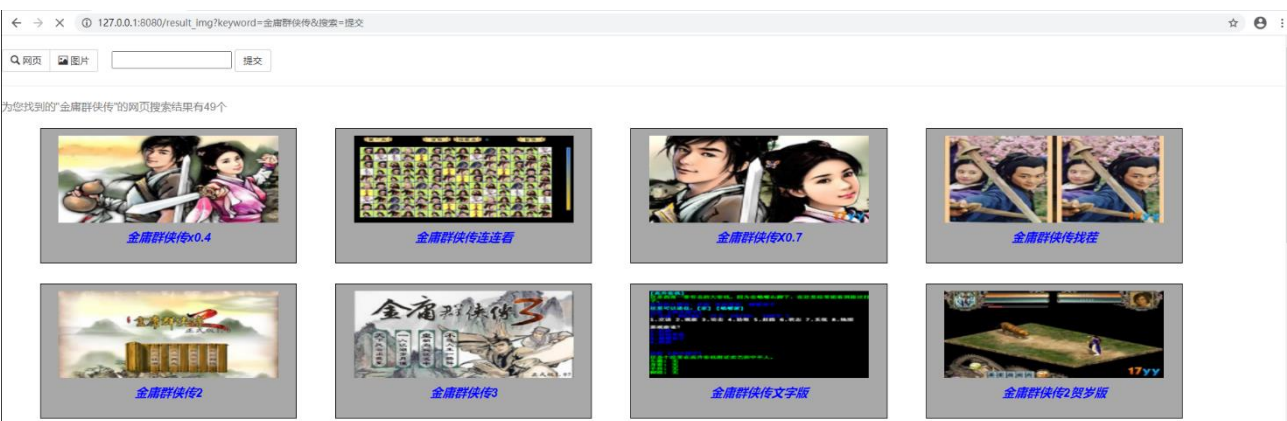
为您找到的“金庸群侠传”的网页搜索结果有994个



在“图片”输入为闪电：



在“图片”输入为金庸群侠传：



六、实验体会

这是本课程的中期实验，转眼便过去了半个学期，而这半个学期我的收获非常丰富。

在本次实验中，由于需要运用之前实验的内容，我再次更新并修正了之前实验中的不足（例如在 lab5 中的存储方式我 data 本来仅负责分词，并不负责存储访问，而为了适配本次实验，我重新调整了一下 data 的存储方式: `t2.setTokenized([False]->[True])`），从而更加贴近实际，实验更加有意义。而且在近几次的融会贯通中，我对相关代码的理解越来越深刻，更能理解其中的内涵。

当然，本次实验也存在不足之处，主要是页面设置的配色等较差，很大原因是我还不懂得如何去查找相应的模板，自己设计模板需要耗费较大的经历和时间，且本次实验对此并不侧重，故我并没有继续完善、美化我的这个搜索网站。