

实验七报告

唐灵

519030910052

F1903002

2020 年 11 月 12 日

摘要

这是电工导 C 课程的第六次实验

1 实验概览

本次实验是一次整合，整合前半期到内容，实现建立图片以及网页的索引，并通过 flask 网页框架进行实现。最后一步是添加 css 样式，让我们的结果更加的好看（虽然最后实际上是变得更丑了）

由于对于其他部分的实现，在以前的报告已经讲的非常清楚了，本次报告主要着重于对于网页部分的实现的表述。

2 实验环境

[illegible]

图 1: 实验运行环境

本次作业文本检索部分使用了 whoosh，具体原因已经在实验六中进行过说明，具体原因不再重复。

其他部分和课程要求内容完全一致，在本地环境中使用 anaconda3 完成实验，python 的版本为 3.8.5，具体环境环境如上图。

3 练习解决思路

3.1 总体思路

总体的实现思路大致可以描述如下：

1. 服务器初始化根目录为搜索页面，通过 search.html 渲染模版。
2. 程序猿从浏览器发起搜索，通过搜索页面向服务器发送请求，通过重定向到结果页面（由于页面设置，结果页面本身也是搜索页面），请求中后缀有”keyword” 参数。
3. 服务器通过获得 keyword 参数，再将 keyword 传入到 search() 函数，并返回搜索结果，这个函数来自于搜索模块，这个模块会对之前创建的索引进行搜索，并返回所有的搜索结果，这个结果足够覆盖结果页面中我们用到的所有信息，并且已经进行过高亮处理。
4. 将返回的搜索结果作为参数传入进网页模版 result.html 进行渲染，得到我们的搜索结果，再传回给浏览器。

对于图片和网页的索引，在这个层面来讲没有任何不同，所以在上述过程中省略掉在源文件中出现的 doc 和 img 后缀。

3.2 网页框架实现思路

具体思路基本上按照以上的逻辑构建，截图如下：

```
def route('/', method='GET'):
    def search_doc():
        return render_template("search_doc.html")
    def route('/w', method='GET'):
        def search_img():
            return render_template("search_img.html")
    def route('/result', method='GET'):
        def result_doc():
            keyword = request.args.get('keyword')
            results = search_doc(keyword)
            return render_template("result_doc.html", results=results, keyword=keyword)
        def result_img():
            keyword = request.args.get('keyword')
            results = search_img(keyword)
            return render_template("result_img.html", results=results, keyword=keyword)
```

图 2: 网页逻辑

3.3 网页模版实现思路

网页模版要接受搜索结果传来的列表类型参数，然后进行循环打印输出，相较于上一次的不同，本次使用了 css 使我们的页面更加的好看，

由于网页结构还是较为简单，选择选用内嵌，与讲 css 样式写进 head 的两种方式来得到我们希望的效果，对于图片，使用 max-width 等方式来使得页面显示能够随比例的变化而变化。

```
<head>
  <title>Result</title>
  <style type="text/css">
    div[class="item"]{
      height: 200px;
      width:20%;
      margin-left: 4%;
      margin-top: 20px;
      background-color: #169850;
      word-wrap: break-word;
      float: left;
    }
    div[class="line"]{
      height:200px;
      width: 100%;
      margin-top: 20px;
      word-wrap: break-word;
    }
    div[class="top"]{
      height: 100px;
      line-height: 100px;
      width: 100%;
      font-size: 40px;
      border: 3px solid #000000;
      text-align: center;
      background-color: #f0f0f0;
    }
    img[class="content"]{
      width: auto;
      height:auto;
      max-width: 75%;
      max-height: 75%;
    }
  </style>
</head>
```

图 3: css 样式设计

4 代码运行结果

最后显示一些搜索结果。

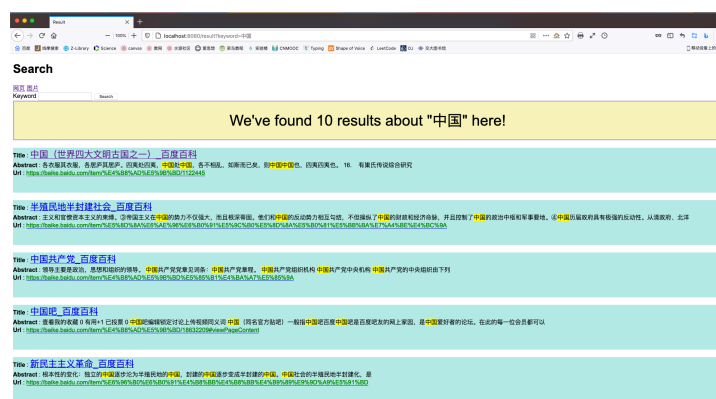


图 4: 网页文档搜索结果

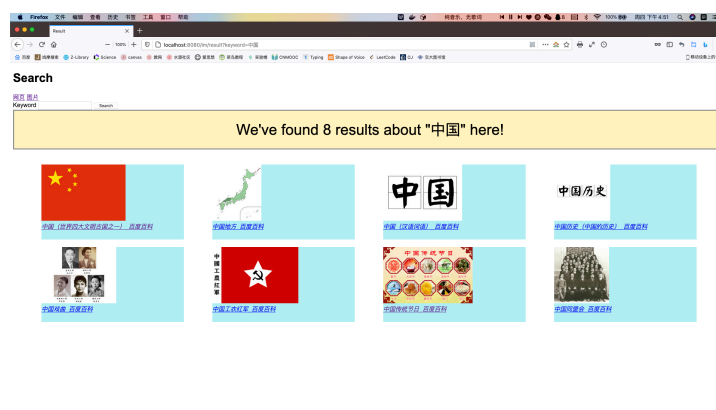


图 5: 图片搜索结果