上机实验六 & 七
实验报告

刘瀚文
517030910294

2018 年 11 月 7 日

目录

Part I

实验六

Chapter 1

实验准备

1.1 环境配置

1.1.1 web.py 安装

安装 web.py

使用 PPT 的方法安装 web.py。

1.2 背景知识

1.2.1 Web 框架

Web 开发

前端网页后端数据库 Query & Request

1.2.2 框架

framework

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。

好处

减少重复开发工作量、缩短开发时间、降低开发成本。同时还有其它的好处，如：使程序设计更合理、程序运行更稳定等。

框架类型

重量级 Zope

中量级 Django Quixote

轻量级 Pylons TurboGears

迷你级 Tornado web.py Bottle & Flask

其他 web2py、uliweb、Karrigell、Werkzeug …

1.3 基础应用

1.3.1 web.py————例程

对于一个站点来说，URL 的组织是最重要的一个部分，因为这是用户看得到的部分，而且直接影响到站点是如何工作的，在著名的站点如：del.icio.us，其 URLs 甚至是网页界面的一部分。而 web.py 以简单的方式就能够构造出一个强大的 URLs。

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

第一部分’/’)是一个匹配 URL 的正则表达式，像/, /help/faq, /item/(+)，等等；第二部分（’ index’）是一个类名，匹配的请求将会被发送过去。

用户通过 URLs（例如 / 或者 /foo?f=1）来请求 web 服务器完成一定请求（例如 GET 或者 POST）。

GET 是最普遍的方法，用来请求一个页面。 当我们在浏览器里输入“harvard.edu”的时候，实际上它是向 Web 服务器请求 GET ”/“。另一个常见的方法是 POST，常用于提交特定类型的表单，例如利用信用卡付费和处理一个订单。

```
class index:
     def GET(self):
      print "Hello, world! “
```

当接收到一个 GET 请求时，上面的 GET 方法将会被 web.py 调用。

```
 if __name__ == "__main__":
   app = web.application(urls, globals())
   app.run()
```

上面告诉 web.py 如何配置 URLs，以及找寻的类在文件中的全局命名空间。

一个完整的 web.py 应用完成，保存为 hello.py

1.3.2 web.py——调用

通过浏览器访问 http://localhost:8080/，会见到命令行或 IDLE 中打印”hello world”。

1.3.3 web.py——模板

在 Python 里面编写 HTML 代码是相当累赘的，而在 HTML 里嵌入 Python 代码则有趣得多。幸运地，web.py 使这过程变得相当容易。

1.3.4 web.py——表单

web.py 的 form 模块可进行建立 html 表单，得到用户输入，验证、导入数据库等操作。Form 模块为 2 个类：Form class 和 Input class Input class 下属子类包括：Textbox, Password, Textarea, Dropdown, Radio, Checkbox, Button。

Chapter 2

实验过程

2.1 实验需求

2.1.1 建立一个简单的搜索引擎

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。

2.2 实验过程

2.2.1 爬取网页————pachong.py

还是使用原来的爬虫 py 文件，但是这次新爬取了携程旅行的网站，同时使用‘ctrip’域名判断，进行对 ctrip 域名的限定查询。

同时在这两周的实践中又总结出许多可以精确化、特异性查找的方法，将在报告的最后展现。

2.2.2 重写网页搜索程序————lab7Search

这是最初版本的搜索实践，使用一个 py 文件，综合了原本的两个图片和文本搜索程序 lab7Search_pic & lab7Search_txt。

同时在搜索程序中由于每次只调用一个函数，所以可以在其中修改传入的索引文件夹名称来修改所用不同索引的位置。这项操作也在报告的最后展示自己升级的新版本操作。

这部分的要点主要是函数的书写，两个函数调用不同的 run() 函数，以进行搜索。同时注意到助教老师在 ppt 中所说明的需要注意的只执行一次 java 的初始化函数。

2.2.3 template

这部分可谓是 practice 的亮点，在 web.py 的使用中，可以将我们的搜索结果呈现到网页中。

我的 template 文件夹主要包括搜索前的 form.html 和展示搜索结果的 result_search.html 文件。

form.html 在这部分中主要是网页的基本结构 + 超链接从 txt 搜索到 pic 搜索的转换

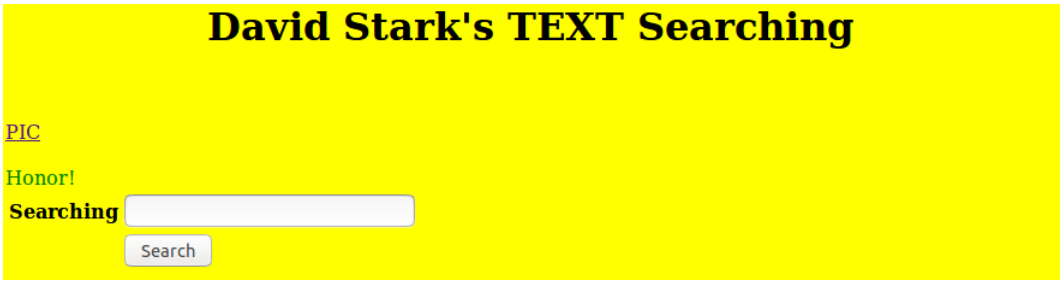
```
$def with(form)
<body bgcolor="lightgreen"></body>
<h1 align="center">David Stark's PIC Searching</h1><br><br>

<a href="/">TEXT</a><br><br>
<font face="verdana" color="red">Honor!</font>
<form action="/i" method="GET">
$:form.render()
</form>
```

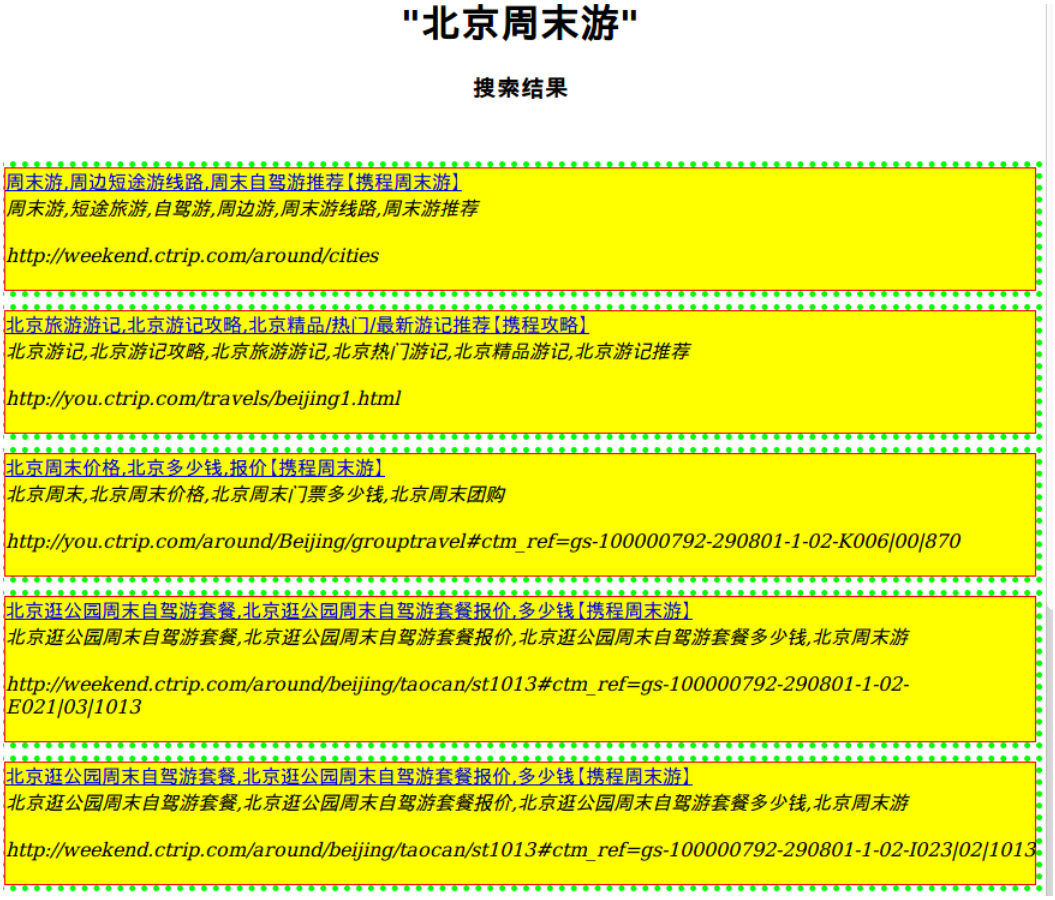
result_search.html 这部分的要点主要是对图片的展现。使用 img 标签进行展示图片。

```
$if name:
$for i in name:
<p>
<img src="$i['imgurl']" />
<a href="$i['url']">$i['title']</a><br>
</p>
```

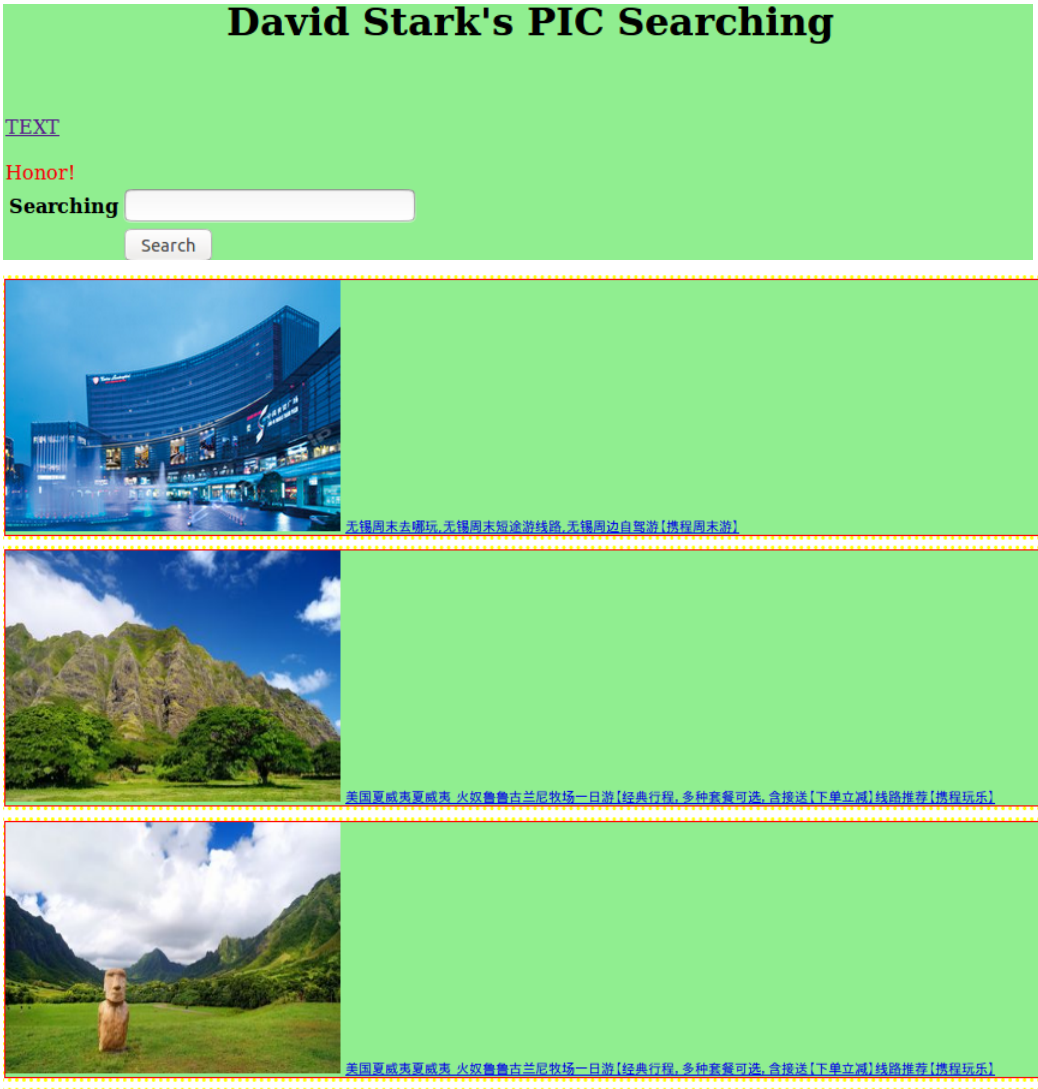
这部分也比较简单

2.2.4 展示搜索的网页





文本搜索 我使用的是黄色，明显而醒目。



图片搜索　我使用的是绿色，清新而护眼。

Part II

实验七

Chapter 3

实验准备

3.1 背景知识

3.1.1 网页页面布局与样式

样式决定了网页中的元素以什么样的形式被显示出来。

可以通过 html 标签属性更改样式。但是有以下两个缺点：写在 html 正文段落中，不利于维护与修改若有多个段落使用同样样式，需要一一修改属性

3.1.2 CSS

CSS 指层叠样式表 (Cascading Style Sheets)

HTML 标签原本被设计为用于定义文档内容。通过使用 <h1>、<p>、<table> 这样的标签，HTML 的初衷是表达“这是标题”、“这是段落”、“这是表格”之类的信息。同时文档布局由浏览器来完成，而不使用任何的格式化标签。

由于两种主要的浏览器（Netscape 和 Internet Explorer）不断地将新的 HTML 标签和属性（比如字体标签和颜色属性）添加到 HTML 规范中，创建文档内容清晰地独立于文档表现层的站点变得越来越困难。

为了解决这个问题，万维网联盟（W3C），这个非营利的标准化联盟，肩负起了 HTML 标准化的使命，并在 HTML 4.0 之外创造出样式（Style）。

把样式添加到 HTML 4.0 中，是为了解决内容与表现分离的问题。

div+css

DIV+CSS(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>、 等 html 标签一样。

CSS 用于创建网页表现（样式/美化）

通过 css 来设置 div 标签样式，这一切常常我们称之为 div+css。

CSS 语法

CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明。

selector declaration1; declaration2; ... declarationN

每条声明由一个属性和一个值组成。

selector property: value

CSS 选择器

ID 选择器 id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式。id 选择器以”#” 来定义。

类选择器 类选择器以一个点号。

属性选择器 对带有指定属性的 HTML 元素设置样式。

创建 CSS

当读到一个样式表时，浏览器会根据它来格式化 HTML 文档。插入样式表的方法有三种

外部样式表 从外部文件读入，可在不同文件中重复使用。

内部样式表 当单个文档需要特殊的样式时，就应该使用内部样式表。（推荐使用）

内联样式

Chapter 4

实验过程

4.1 实验需求

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

4.2 中期整合

4.2.1 form.html

渲染使网页呈现不同的颜色，txt 使用的是黄颜色，pic 使用的是绿颜色。

formtxt 这部分使用的是简单的样式，直接在标签内部进行属性的设置。

```
<body bgcolor="yellow"></body>

<h1 align="center">David Stark's TEXT Searching</h1><br><br>
<font face="verdana" color="green">Honor!</font>
```

formtxt 具有相似的结构。

4.2.2 result_search.html

这一部分在头文件部分进行描述，建立对整个 html 都有效的 css 格式设置。

result_search_txt 设置边框的样式及外部点状团的形式。

```
<head>
<style type="text/css">
p
{
  background-color: yellow;
  border:red solid thin;
  outline:#00ff00 dotted 5px;
}
h1{
  text-align :center;
}
h3{
  text-align :center;
}
</style>
</head>
```

result_search_img 具有相似的结构。

Part III

My EXTRA
图片搜索引擎

最近由于写搜索引擎（电工导）产生了极大的兴趣，导致每天起床之后就想爬网站，建立属于自己的搜索引擎。在这个前提之下，我在完成基础实验的情况下，还对已有代码进行了完善，同时建立了一个可以同时进行风景、美食、汽车、女孩子的图片搜索引擎。

在接下来的部分，将展示我的研究成果。

Chapter 5

完善的代码

5.1 爬虫文件——savedpachong.py

5.1.1 更加优化的特异性搜索

get_all_links() 在这个函数中，还使用了 decode() 操作进行判断，这个也是新学习到的点。

额外的，判断了 img 父级的标签，这样可以少爬取重复的网站。

```
def get_all_links(content, page):
  links = []
  soup = BeautifulSoup(content, features="html.parser")
  for i in soup.findAll('a', {'href':     re.compile('^http|^/')}):
    url = i['href']

    if 'fengjingsheying/2018'.decode() in url:

      if 'li'.decode() in i.parent.name:

        newUrl = urlparse.urljoin(page, url)
        links.append(newUrl)

  return links
```

5.1.2 搜索时的双重认证

```
    if 'fengjingsheying' in page:
      content = get_page(page)
```

5.2 创建图片索引——IndexPicFile.py

5.2.1 getinFo(img)

获取图片信息的时候，对于所爬取得网站进行了特异性优化，使得爬取的时候肯定能够爬取到所需要得网页信息。

5.2.2 description

由于图片的性质，我认为在搜索引擎中加入图片的描述信息比较好，所以同时还获取的 description 信息，并将其加入索引。

5.3 搜索程序——lab7_Search.py

5.3.1 精简了搜索程序

只需要一个搜索函数，就可以完成所有需要的搜索要求。

在搜索 py 文件中，不再指定对应的索引文件位置，而是在主函数调用的时候赋予同一个图片搜索引擎不同的索引位置。

5.4 主程序——practice.py

5.4.1 完善了综合搜索能力

建立较为完善的名称和综合索引能力，使得可以同时搜索四种不同类型的图片。

风景摄影 照片搜索

```
'/', 'index_fengjingsheying',
'/fengjingsheying', 'fengjingsheying_image',
```

美食图片　照片搜索

```
'/i_meishitupian', 'index_meishitupian',
'/meishitupian', 'meishi_image'
```

汽车　照片搜索

```
'/i_qiche', 'index_qiche',
'/qiche', 'qiche_image',  # wudihaohua
```

女孩子　照片搜索

```
'/i_meinv', 'index_meinv',
'/meinv', 'meinv_image',  # meizi
```

5.4.2　重新写了搜索函数

重写的函数使得搜索可以在转换搜索引擎的时候切换搜索所用索引的位置，而避免在搜索部分使用多个函数，减少了代码的冗余，精简了代码。

5.5　呈现部分——template

以下的展示以 formimg.html 和 formfengjingsheying.html 为例展示。

5.5.1　formimg.html

```
$def with(form)

<body bgcolor="#808080"></body>

<h1 align="center">David Stark's QiChe Searching</h1><br><br>
<a href="/i_meinv">PIC_MEINV</a><br><br>


<form action="/qiche" method="GET">
$:form.render()
</form>
```

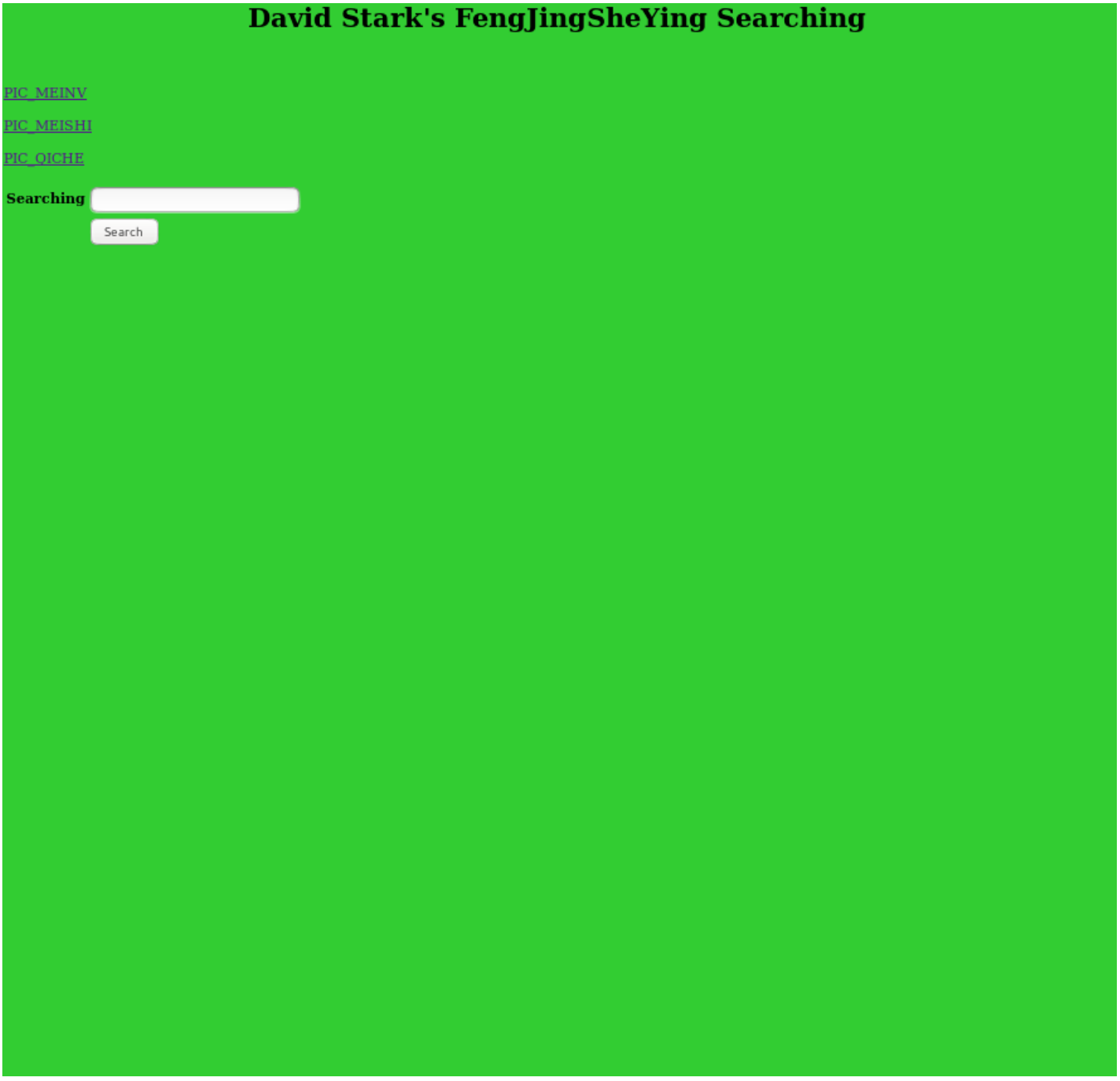
5.5.2 formfengjingsheying.html

```
$def with(form)

<body bgcolor="#32cd32"></body>

<h1 align="center">David Stark's FengJingSheYing Searching</h1><br><br>

<a href="/i_meinv">PIC_MEINV</a><br><br>
<a href="/i_meishitupian">PIC_MEISHI</a><br><br>
<a href="/i_qiche">PIC_QICHE</a><br><br>

<form action="/fengjingsheying" method="GET">
$:form.render()
</form>
```

Chapter 6

搜索引擎的展示

展示所创造的页面，由于这个搜索引擎侧重于功能上的实现，所以在看不见的代码部分进行了充分的优化，使得无论从爬取的网页的角度，还是搜索时呈现的内容的角度，都有一个显著的提高。由于侧重于内部（或者说后端的建设），前端美化的部分是用颜色区分不同的搜索引擎。


David Stark's FengJingSheYing Searching
PIC_MEINV
PIC_MEISHI
PIC_QICHE
Searching
Search

风景摄影的搜索界面

click here 自然大瀑布

自然大瀑布大自然的美妙之处就在于它的美丽，而美丽的风景往往和花花草草有着非常紧密的联系，喜欢淡雅植物是很多人的天性。当你看着眼前缤纷绚丽，迷人的景色时，你是否感受到了温馨呢？看着这样的美景，是多么享受的一件事情，一组自然大瀑布分享，我们是不是已经深深的融入了呢？喜欢的朋友不妨记得收藏本站哦。

click here 清新自然大自然的色彩桌面壁纸

雪花在寂静的路上飘零，凝结成冰块，冰天雪地之中冬天来了！清新自然大自然的色彩桌面壁纸。今天小编就与大家分享一组清新自然大自然的色彩桌面壁纸，供大家选择。

风景摄影的搜索结果


David Stark's MeiShiTuPian Searching
PIC_FENGJINGSHEYING
PIC_MEINV
PIC_QICHE
Searching
Search

美食图片的搜索界面

click here 辣椒水果素材

让人垂涎欲滴的美食。一组辣椒水果素材分享，如果味蕾不能亲自感受，那心情就像冬天的萝卜、心寒又心空。如果你对美食也颇有一番研究，那就不妨跟随小编一起来欣赏这组辣椒水果素材，希望大家会喜欢。

click here 蓝莓水果冰淇淋

我们很喜欢把生活的苦涩藏心头，把幸福寄托在美食，呈现在每一桌的餐桌上，一组蓝莓水果冰淇淋小编分享给大家，希望大家喜欢。怎么样，听完了小编的介绍，对美食有了一定的了解吧，下面就一起来欣赏这组蓝莓水果冰淇淋吧。

美食图片的搜索结果


David Stark's QiChe Searching
PIC_FENGJINGSHEYING
PIC_MEINV
PIC_MEISHI
Searching
无敌豪华
Search

汽车的搜索界面

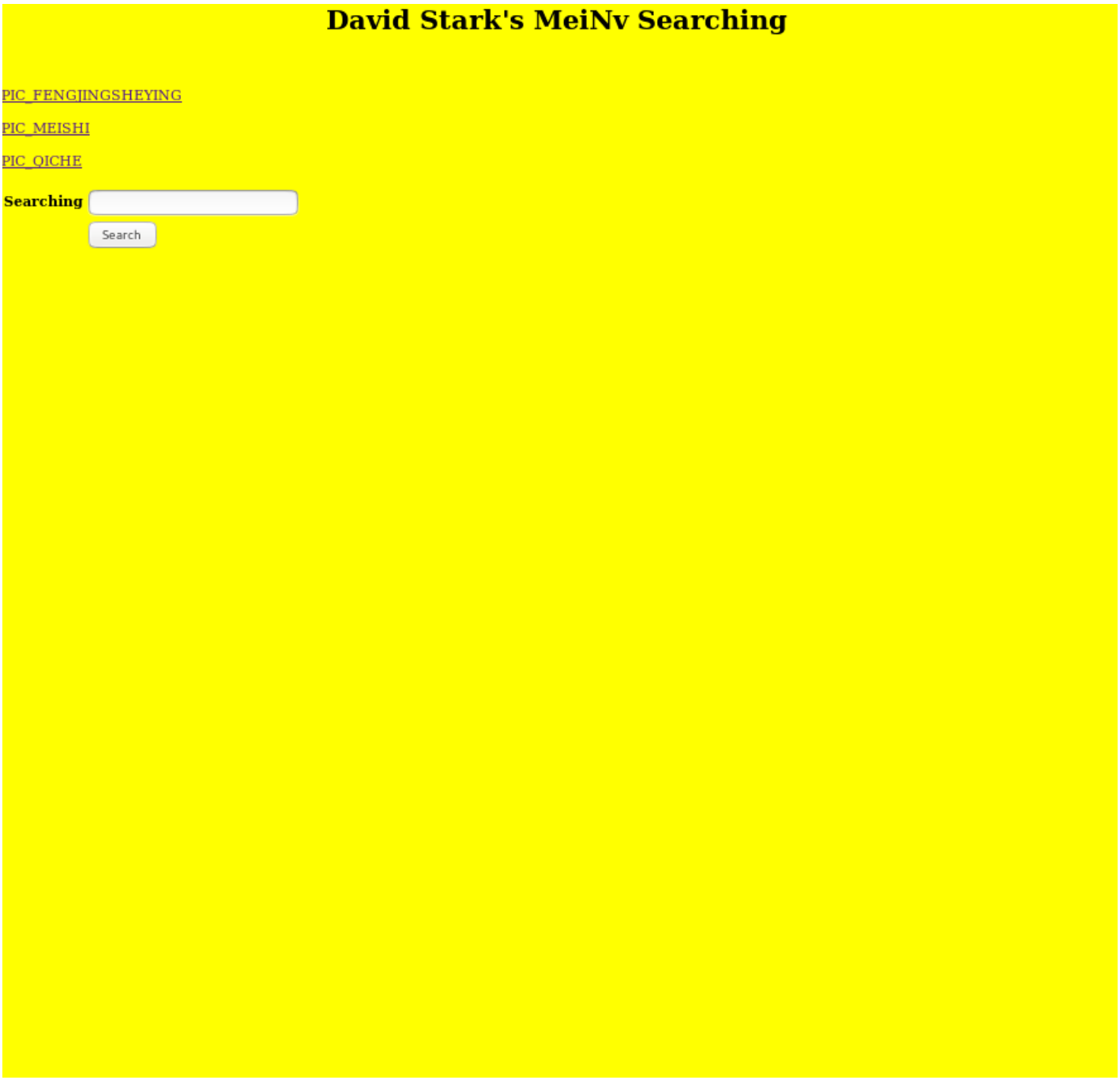
click here 豪华汽车凯迪拉克高清图片

豪华汽车凯迪拉克代表着最高品质和形象，如红色表示勇猛和赤胆；银色表示婚姻、纯洁、博爱和美德；黄色表示丰收和富有；蓝色表示创新和探险。

click here 高端时尚大气的宝马豪车图片

分享一组高端时尚大气的宝马豪车图片，宝马德系三大豪华品牌之一，宝马的车系有1、2、3、4、5、6、7、i、X、Z、等几个系列。

汽车的搜索结果



女孩子的搜索界面　　搜索结果界面暂时保密

Part IV

实验总结

本次实验在之前实验的基础上，进行研究和拓展。实验六进行了网页上的初步尝试，也是我们的搜索结果可视化的重要步骤，也正是这一次实验，激发了我对于写搜索引擎巨大的兴趣，也是让我这几天一直都很兴奋的爬各种网站进行优化的动力。实验七主要是使用 css 进行对页面的美化，我也初步进行了尝试。还学习了翻页的操作，但是由于目前的使用结果没有太大的技术含量就暂时没放到程序中。

实验中的基础操作并不是很难，但是在不断想如何优化，尝试理解每一行写过的代码以及其背后每一个返回结果的时候用了大量的时间反复运行，也不断的进行了网络爬虫实验。我希望这段时间的大量努力，也能为之后的操作和实验打下良好的基础吧。

继续努力，不断进步！争取在实践中获得更多的知识！