



## 第3章 Python语言入门与Web信息解析

---

3.1 Python语言入门

3.2 Python语言进阶

3.3 HTTP解析与Python实现

3.4 HTML解析与Python实现

3.5 Web信息解析



## 3.4 HTML解析与Python实现

---

### 3.4.1 HTML与CSS要素

### 3.4.2 BeautifulSoup库及对象

### 3.4.3 BeautifulSoup库遍历文档树

### 3.4.4 BeautifulSoup库搜索文档树

### 3.4.5 BeautifulSoup库查找CSS过滤器

### 3.4.6 Python解析网页实例



## 3.4.1 HTML与CSS要素

---

- Web网页主要由三部分组成：
  - 结构（Structure）
    - 主要包括XHTML和XML，
  - 表现（Presentation）
    - 主要包括CSS
  - 行为（Behavior）。
    - 主要包括对象模型（如W3C DOM）、CMAScript

## 3.4.1 HTML与CSS

### 1. 文档标记

```
<html>
<head>
  <title>Python爬虫开发与项
  <meta charset="UTF-8">
</head>
<body>
  文档设置标记<br>
  <p>这是段落。</p>
  <p>这是段落。</p>
  <p>这是段落。</p>
  <hr>
  <center>居中标记1</center>
  <center>居中标记2</center>
  <hr>
```

```
<ul>
  <li>Coffee</li>
  <li>Milk</li>
</ul>
<ol type="A">
  <li>Coffee</li>
  <li>Milk</li>
</ol>
  <dl>
    <dt>计算机</dt>
    <dd>用来计算的仪器 ... ..</dd>
    <dt>显示器</dt>
    <dd>以视觉方式显示信息的装置 ... ..</dd>
  </dl>
  <div >
    <h3>这是标题</h3>
    <p>这是段落。</p>
  </div>
```

```
<pre>
[00:00](music)
[00:28]你我皆凡人, 生在人世间;
[00:35]终日奔波苦, 一刻不得闲;
[00:43]既然不是仙, 难免有杂念;
</pre>
<hr>
<p>
[00:00](music)
[00:28]你我皆凡人, 生在人世间;
[00:35]终日奔波苦, 一刻不得闲;
```

## 3.4.1 HTML与CSS要素

### 2. 图像标记<img>



#### ■ alt属性有三个作用：

- 1) 当网页上的图片被加载完成后，鼠标移动到上面去，会显示这个图片指定的属性文字；
- 2) 如果图像没有下载或者加载失败，会用文字来代替图像显示；
- 3) 搜索引擎可以通过这个属性的文字来抓取图片

```
47         <h1>
48             <a href="https://www.cnblogs.com/" title="开发者的网上家园">
</a>
49         </h1>
50     </div>
```



## 3.4.1 HTML与CSS要素

### 3.超链接标记

- 爬虫开发中经常需要抽取链接，链接的引用使用的是<a>标记。

- <a>标记的基本语法：

<a href=“链接地址” target=“打开方式” name=“页面锚点名称”>链接文字或者图片</a>

- name属性用来指定页面的锚点名称。

- \_blank时：在一个新的窗口中打开链接；
- \_self（默认值）：在当前窗口中打开链接
- ...



## 3.4.1 HTML与CSS要素

### 4. 表格标记

基本结构包括<table>、<caption>、<tr>、<td>和<th>等

```
<table width="960" align="center" border="1" rules="all" cellpadding="15">
```

```
<tr>
    <th>学号</th>
    <th>班级</th>
    <th>姓名</th>
    <th>年龄</th>
    <th>籍贯</th>

</tr>
<tr align="center">
    <td>1500001</td>
    <td>(1)班</td>
    <td>张三</td>
    <td>16</td>
    <td>上海</td>

</tr>
```

## 3.4.1 HTML与CSS要素

### ■ CSS层叠样式表（Cascading Style Sheets）

- 用来定义如何显示HTML元素，一般和HTML配合使用
- 为了解决内容与表现分离的问题，即使同一个HTML文档也能表现出外观的多样化

```
98 <body class="sohu-index-v3" data-spm="home">
99   <header class="sohu-head">
100     <div class="area sohu-head-box">
101       <div class="right head-right">
102
103       </div>
104     </div>
105   </header><div class="sohu-ph" id="sohuTopc" style="display:none;">
106     <div class="ph-link">
107       <a href="http://news.sohu.com/s2018/guoqing-2/index.shtml" target="blank"></a>
108     </div>
109     <div id="ph-close" class="ph-close"><a href="javascript:">
110   </div>   <!-- 皮肤 wrapper box -->
111     <div class="theme-skin-wrap" data-spm="top-festival">
112       <div class="mask"></div>
```

class:列出一系列以空格分隔的CSS类名

id:将页面唯一标识符附加到某个标签

使用id可以快速获取我们感兴趣的HTML页面的某些部分





## 3.4.1 HTML与CSS要素

---

### ■ CSS格式

- 样式信息被写为冒号分隔的基于键值的语句列表
- 每个语句本身用分号隔开

```
color: 'red';  
background-color: #ccc;  
font-size: 14pt;  
border: 2px solid yellow;
```

## 3.4.1 HTML与CSS要素

### ■ CSS样式声明有三种方式：

#### ■ 内联样式表：

- CSS代码直接写在现有的HTML标记中，使用style属性改变样式  
`<body style="background-color:green; margin:0; padding:0;"></body>`

#### ■ 嵌入式样式表：

样式应用在哪个标签很清晰

- CSS样式代码写在<style type="text/css"></style>标记之间
- 嵌入式CSS样式写在<head></head>之间。

#### ■ 外部样式表：

- CSS代码写一个单独的外部文件中，以“.css”为扩展名
- 在<head>内（不是在<style>标记内）使用<link>标记将CSS样式文件链接到HTML文件内
- `<link rel="StyleSheet" type="text/css" href="style.css">`。

需要定义选择器

## ■ CSS规则由两个主要的部分构成：

```
selector {declaration1; declaration2; ... declarationN }
```

### ■ 选择器：

- 需要改变样式的HTML元素

- 三种选择器：标签选择器、类选择器、ID选择器

可以任意组合

### ■ 一条或多条声明：

- 用于定义显示方式，每条声明由一个属性和一个值组成
- 属性（**property**）是希望设置的样式属性（**style attribute**）。每个属性有一个值。属性和值由冒号分开
- 例如：将h1标记中的颜色设置为红色，字体大小为14px



```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>实例</title>
    <style type='text/css'>
```

```
    p, h3{
      color:red;设置文本颜色
    }
  </style>
```

标签选择器的使用

```
</head>
<body>
  <h3>啦啦啦</h3>
  <p>你好</p>
  我的朋友</p>
</body>
</html>
```

样式（color:red;）将应用到p,h3这两个选择器所引用的元素。

## ■ 标签选择器

- 选择具有特定标签名称的所有元素
- 如<p>、<h3>
- 代码：

```
p{font-size:12px;line-height:1.6em;}
```

啦啦啦

你好

我的朋友

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>实例</title>
  <style>
    .box {
      color: cyan;
      font-size: 40px;
    }
  </style>
</head>
<body>
  <div class="box">
    <span>盒子里有</span>
    <span>小动物</span>
    <p>小猫咪</p>
    <p>小跳蛙</p>
    <p id="t">小老虎</p>
  </div>
</body>
</html>

```

## ■ 类选择器

- 选择HTML中定义的特定类的所有元素，即class出现的位置
- 语法：
  - .类名称{css样式代码;}

盒子里有  
小动物  
小猫咪  
小跳蛙  
小老虎

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>实例</title>
    <style type='text/css'>
      #tale{
        color: aqua;
      }
      #tiger{
        color: red;
      }
    </style>
  </head>
  <body>
    <p>
      <span id="tale">一二三四五 上
    </p>
  </body>
</html>

```

id选择器的使用

一二三四五 上山打老虎

## ■ id选择器

- 按照id属性值选择匹配的元素
- 任何的标签都可以设置id
- 正确的HTML文档应该确保每个"id"唯一，并且只给与一个元素
- 语法：  
#intro {font-weight:bold;}



- Python可以使用相同的CSS选择器语法快速查找和检索HTML页面中的元素
- 例：使用开发者工具(chrome)Elements，选中某元素，点击右键选择copy→copy selector，可以获得CSS选择器

