**1.获取<tr></tr>标签之间内容**

该部分主要是通过正则表达式获取两个标签之间的内容，通常这种标签都是成对出现的。  
开始标签如：<tr>、<th>、<td>、<a>、<table>、<div>...  
后缀标签如：</tr>、</th>、</td>、</a>、</table>、</div>...  
  
核心代码：  
        **res\_tr = r'<tr>(.\*?)</tr>'  
        m\_tr =  re.findall(res\_tr,language,re.S|re.M)**  
例子：

1. *# coding=utf-8*
2. import re
4. language = '''<tr><th>性別：</th><td>男</td></tr><tr>'''
6. *#正则表达式获取<tr></tr>之间内容*
7. res\_tr = r'<tr>(.\*?)</tr>'
8. m\_tr = re.findall(res\_tr,language,re.S|re.M)
9. for line in m\_tr:
10. print line
11. *#获取表格第一列th 属性*
12. res\_th = r'<th>(.\*?)</th>'
13. m\_th = re.findall(res\_th,line,re.S|re.M)
14. for mm in m\_th:
15. print unicode(mm,'utf-8'), *#unicode防止乱*
16. *#获取表格第二列td 属性值*
17. res\_td = r'<td>(.\*?)</td>'
18. m\_td = re.findall(res\_td,line,re.S|re.M)
19. for nn in m\_td:
20. print unicode(nn,'utf-8')

输出如下所示：

1. >>>
2. <th>性別：</th><td>男</td>
3. 性別： 男
4. >>>

python通过re模块提供对正则表达式的支持。使用re的一般步骤是先将正则表达式的字符串形式编译为Pattern实例，然后使用Pattern实例处理文本并获得匹配结果（一个Match实例），最后使用Match实例获得信息，进行其他的操作。  
  
findall(string[, pos[, endpos]]) | re.findall(pattern, string[, flags]): 搜索string，以列表形式返回全部能匹配的子串。其中RE的常见参数包括：  
        re.I(re.IGNORECASE): 忽略大小写（括号内是完整写法）  
        re.M(re.MULTILINE): 多行模式，改变'^'和'$'的行为  
        re.S(re.DOTALL): 点任意匹配模式，改变'.'的行为

------------------------------------------------------------------------------------------------------------------------------

**2.获取超链接<a href=..></a>之间内容**

通常在使用正则表达式时，需要分析网页链接，获取URL或网页内容。核心代码如下：  
        **res = r'<a .\*?>(.\*?)</a>'  
        mm =  re.findall(res, content, re.S|re.M)  
        urls=re.findall(r"<a.\*?href=.\*?<\/a>", content, re.I|re.S|re.M)**例子：

1. *# coding=utf-8*
2. import re
4. content = '''
5. <td>
6. <a href="https://www.baidu.com/articles/zj.html" title="浙江省">浙江省主题介绍</a>
7. <a href="https://www.baidu.com//articles/gz.html" title="贵州省">贵州省主题介绍</a>
8. </td>
9. '''
11. *#获取<a href></a>之间的内容*
12. print u'获取链接文本内容:'
13. res = r'<a .\*?>(.\*?)</a>'
14. mm =  re.findall(
15. res, content, re.S|re.M)
16. for value in mm:
17. print value
19. *#获取所有<a href></a>链接所有内容*
20. print u'\n获取完整链接内容:'
21. urls=re.findall(r"<a.\*?href=.\*?<\/a>", content, re.I|re.S|re.M)
22. for i in urls:
23. print i
25. *#获取<a href></a>中的URL*
26. print u'\n获取链接中URL:'
27. res\_url = r"(?<=href=\").+?(?=\")|(?<=href=\').+?(?=\')"
28. link = re.findall(res\_url ,  content, re.I|re.S|re.M)
29. for url in link:
30. print url

输出如下图所示：

1. >>>
2. 获取链接文本内容:
3. 浙江省主题介绍
4. 贵州省主题介绍
6. 获取完整链接内容:
7. <a href="https://www.baidu.com/articles/zj.html" title="浙江省">浙江省主题介绍</a>
8. <a href="https://www.baidu.com//articles/gz.html" title="贵州省">贵州省主题介绍</a>
10. 获取链接中URL:
11. https://www.baidu.com/articles/zj.html
12. https://www.baidu.com//articles/gz.html
13. >>>

当然如果是通过Selenium分析DOM树结构获取href对应的url或title中的值，其核心代码如下所示，这里主要是给大家做个对比，理解不同方法的优势：  
      **driver.get(link)   
      elem = driver.find\_elements\_by\_xpath("//div[@class='piclist']/tr/dd[1]")     
      for url in elem:      
            pic\_url = url.get\_attribute("href")      
            print pic\_url**  
参考文章：[[python爬虫] Selenium定向爬取虎扑篮球海量精美图片](http://blog.csdn.net/eastmount/article/details/49393483)

------------------------------------------------------------------------------------------------------------------------------

**3.获取URL最后一个参数命名图片或传递参数**

通常在使用Python爬取图片过程中，会遇到图片对应的URL最后一个字段通常用于命名图片，如虎扑孙悦妻子图片：

[http://i1.hoopchina.com.cn/blogfile/201411/11/BbsImg141568417848931\_640\*640.jpg](http://i1.hoopchina.com.cn/blogfile/201411/11/BbsImg141568417848931_640*640.jpg)  
此时需要通过该URL的"/"后面的参数命名图片，则方法如下：

1. urls = "http://i1.hoopchina.com.cn/blogfile/201411/11/BbsImg141568417848931\_640\*640.jpg"
2. values = urls.split('/')[-1]
3. print values

输出如下所示：

1. >>>
2. BbsImg141568417848931\_640\*640.jpg
3. >>>

在使用Python获取GET方法的URL链接中，还可能存在传递参数的值。  
此时获取参数方法如下：

1. url = 'http://localhost/test.py?a=hello&b=world'
2. values = url.split('?')[-1]
3. print values
4. for key\_value in values.split('&'):
5. print key\_value.split('=')

输出如下所示：

1. >>>
2. a=hello&b=world
3. ['a', 'hello']
4. ['b', 'world']
5. >>>

------------------------------------------------------------------------------------------------------------------------------  
  
**4.爬取网页中所有URL链接**

在学习爬虫过程中，你肯定需要从固有网页中爬取URL链接，再进行下一步的循环爬取或URL抓取。如下，爬取CSDN首页的所有URL链接。

1. *# coding=utf-8*
2. import re
3. import urllib
5. url = "http://www.csdn.net/"
6. content = urllib.urlopen(url).read()
7. urls = re.findall(r"<a.\*?href=.\*?<\/a>", content, re.I)
8. for url in urls:
9. print unicode(url,'utf-8')
11. link\_list = re.findall(r"(?<=href=\").+?(?=\")|(?<=href=\').+?(?=\')", content)
12. for url in link\_list:
13. print url

输出如下所示：

1. >>>
2. <a href="https://passport.csdn.net/account/login?from=http://my.csdn.net/my/mycsdn">登录</a>
3. <a href="http://passport.csdn.net/account/mobileregister?action=mobileRegister">注册</a>
4. <a href="https://passport.csdn.net/help/faq">帮助</a>
5. ...
6. https://passport.csdn.net/account/login?from=http://my.csdn.net/my/mycsdn
7. http://passport.csdn.net/account/mobileregister?action=mobileRegister
8. https://passport.csdn.net/help/faq
9. ...
10. >>>

------------------------------------------------------------------------------------------------------------------------------  
  
**5.爬取网页标题title两种方法**  
  
获取网页标题也是一种常见的爬虫，如我在爬取维基百科国家信息时，就需要爬取网页title。通常位于<html><head><title>标题</title></head></html>中。  
下面是爬取CSDN标题的两种方法介绍：

1. *# coding=utf-8*
2. import re
3. import urllib
5. url = "http://www.csdn.net/"
6. content = urllib.urlopen(url).read()
8. print u'方法一:'
9. title\_pat = r'(?<=<title>).\*?(?=</title>)'
10. title\_ex = re.compile(title\_pat,re.M|re.S)
11. title\_obj = re.search(title\_ex, content)
12. title = title\_obj.group()
13. print title
15. print u'方法二:'
16. title = re.findall(r'<title>(.\*?)</title>', content)
17. print title[0]

输出如下所示：

1. >>>
2. 方法一:
3. CSDN.NET - 全球最大中文IT社区，为IT专业技术人员提供最全面的信息传播和服务平台
4. 方法二:
5. CSDN.NET - 全球最大中文IT社区，为IT专业技术人员提供最全面的信息传播和服务平台
6. >>>

------------------------------------------------------------------------------------------------------------------------------  
  
**6.定位table位置并爬取属性-属性值**  
  
如果使用Python库的一些爬取，通常可以通过DOM树结构进行定位，如代码：  
    login = driver.find\_element\_by\_xpath("//form[@id='loginForm']")   
参考文章：[[Python爬虫] Selenium实现自动登录163邮箱和Locating Elements介绍](http://blog.csdn.net/eastmount/article/details/47825633)  
  
但如果是正则表达式这种相对传统傻瓜式的方法，通过通过find函数寻找指定table方法进行定位。如：获取Infobox的table信息。  
通过分析源代码发现“程序设计语言列表”的消息盒如下：  
<table class="infobox vevent" ..><tr><th></th><td></td></tr></table>

1. start = content.find(r'<table class="infobox vevent"') *#起点记录查询位置*
2. end = content.find(r'</table>')
3. infobox = language[start:end]
4. print infobox

print infobox 输出其中一门语言ActionScript的InfoBox消息盒部分源代码如下：

1. <table class="infobox vevent" cellspacing="3" style="border-spacing:3px;width:22em;text-align:left;font-size:small;line-height:1.5em;">
2. <caption class="summary"><b>ActionScript</b></caption>
3. <tr>
4. <th scope="row" style="text-align:left;white-space:nowrap;;;">发行时间</th>
5. <td style=";;">1998年</td>
6. </tr>
7. <tr>
8. <th scope="row" style="text-align:left;white-space:nowrap;;;">实现者</th>
9. <td class="organiser" style=";;"><a href="/wiki/Adobe\_Systems" title="Adobe Systems">Adobe Systems</a></td>
10. </tr>
11. <tr>
12. <tr>
13. <th scope="row" style="text-align:left;white-space:nowrap;;;">启发语言</th>
14. <td style=";;"><a href="/wiki/JavaScript" title="JavaScript">JavaScript</a>、<a href="/wiki/Java" title="Java">Java</a></td>
15. </tr>
16. </table>

参考文章：[[python学习] 简单爬取维基百科程序语言消息盒](http://blog.csdn.net/eastmount/article/details/44342559)  
然后再在这个infobox内容中通过正则表达式进行分析爬取。下面讲述爬取属性-属性值：  
爬取格式如下：  
        <table>  
                <tr>  
                      <th>属性</th>  
                      <td>属性值</td>  
                </tr>  
        </table>  
其中th表示加粗处理，td和th中可能存在属性如title、id、type等值；同时<td></td>之间的内容可能存在<a href=..></a>或<span></span>或<br />等值，都需要处理。下面先讲解正则表达式获取td值的例子：  
参考：<http://bbs.csdn.net/topics/390353859?page=1>

1. <table>
2. <tr>
3. <td>序列号</td><td>DEIN3-39CD3-2093J3</td>
4. <td>日期</td><td>2013年1月22日</td>
5. <td>售价</td><td>392.70 元</td>
6. <td>说明</td><td>仅限5用户使用</td>
7. </tr>
8. </table>

Python代码如下：

1. *# coding=utf-8*
2. import re
4. s = '''<table>
5. <tr>
6. <td>序列号</td><td>DEIN3-39CD3-2093J3</td>
7. <td>日期</td><td>2013年1月22日</td>
8. <td>售价</td><td>392.70 元</td>
9. <td>说明</td><td>仅限5用户使用</td>
10. </tr>
11. </table>
12. '''
14. res = r'<td>(.\*?)</td><td>(.\*?)</td>'
15. m = re.findall(res,s,re.S|re.M)
16. for line in m:
17. print unicode(line[0],'utf-8'),unicode(line[1],'utf-8') *#unicode防止乱码*
19. *#输出结果如下：*
20. *#序列号 DEIN3-39CD3-2093J3*
21. *#日期 2013年1月22日*
22. *#售价 392.70 元*
23. *#说明 仅限5用户使用*

如果<td id="">包含该属性则正则表达式为r'<td id=.\*?>(.\*?)</td>'；同样如果不一定是id属性开头，则可以使用正则表达式r'<td .\*?>(.\*?)</td>'。  
  
  
------------------------------------------------------------------------------------------------------------------------------  
  
**7.过滤<span></span>等标签**  
  
在获取值过程中，通常会存在<span>、<br>、<a href>等标签，下面举个例子过滤。  
<td><span class="nickname">(字) 翔宇</span></td>过滤标签<span>核心代码：  
**elif "span" in nn: #处理标签<span>  
            res\_value = r'<span .\*?>(.\*?)</span>'  
            m\_value = re.findall(res\_value,nn,re.S|re.M)   
            for value in m\_value:  
                print unicode(value,'utf-8'),**  
代码如下，注意print中逗号连接字符串：

1. *# coding=utf-8*
2. import re
4. language = '''
5. <table class="infobox bordered vcard" style="width: 21em; font-size: 89%; text-align: left;" cellpadding="3">
6. <caption style="text-align: center; font-size: larger;" class="fn"><b>周恩来</b></caption>
7. <tr>
8. <th>性別：</th>
9. <td>男</td>d
10. </tr>
11. <tr>
12. <th>異名：</th>
13. <td><span class="nickname">(字) 翔宇</span></td>
14. </tr>
15. <tr>
16. <th>政黨：</th>
17. <td><span class="org"><a href="../articles/%E4%B8%AD9A.html" title="中国共产党">中国共产党</a></span></td>
18. </tr>
19. <tr>
20. <th>籍貫：</th>
21. <td><a href="../articles/%E6%B5%9981.html" title="浙江省">浙江省</a><a href="../articles/%E7%BB%8D82.html" title="绍兴市">绍兴市</a></td>
22. </tr>
23. </table>
24. '''
26. *#获取table中tr值*
27. res\_tr = r'<tr>(.\*?)</tr>'
28. m\_tr = re.findall(res\_tr,language,re.S|re.M)
29. for line in m\_tr:
30. *#获取表格第一列th 属性*
31. res\_th = r'<th>(.\*?)</th>'
32. m\_th = re.findall(res\_th,line,re.S|re.M)
33. for mm in m\_th:
34. if "href" in mm: *#如果获取加粗的th中含超链接则处理*
35. restr = r'<a href=.\*?>(.\*?)</a>'
36. h = re.findall(restr,mm,re.S|re.M)
37. print unicode(h[0],'utf-8'), *#逗号连接属性值 防止换行*
38. else:
39. print unicode(mm,'utf-8'), *#unicode防止乱*
41. *#获取表格第二列td 属性值*
42. res\_td = r'<td>(.\*?)</td>' *#r'<td .\*?>(.\*?)</td>'*
43. m\_td = re.findall(res\_td,line,re.S|re.M)
44. for nn in m\_td:
45. if "href" in nn: *#处理超链接<a href=../rel=..></a>*
46. res\_value = r'<a .\*?>(.\*?)</a>'
47. m\_value = re.findall(res\_value,nn,re.S|re.M)
48. for value in m\_value:
49. print unicode(value,'utf-8'),
50. elif "span" in nn: *#处理标签<span>*
51. res\_value = r'<span .\*?>(.\*?)</span>'
52. m\_value = re.findall(res\_value,nn,re.S|re.M) *#<td><span class="nickname">(字) 翔宇</span></td>*
53. for value in m\_value:
54. print unicode(value,'utf-8'),
55. else:
56. print unicode(nn,'utf-8'),
57. print ' ' *#换行*

输出如下所示：

1. >>>
2. 性別： 男
3. 異名： (字) 翔宇
4. 政黨： 中国共产党
5. 籍貫： 浙江省 绍兴市
6. >>>

------------------------------------------------------------------------------------------------------------------------------  
  
**8.获取<script></script>等标签内容**  
  
比如在获取游讯网图库中，图集对应的原图它是存储在script中，其中获取原图-original即可，缩略图-thumb，大图-big，通过正则表达式下载URL：  
        res\_original = r'"original":"(.\*?)"' #原图  
        m\_original = re.findall(res\_original,script)  
代码如下：

1. *# coding=utf-8*
2. import re
3. import os
5. content = '''
6. <script>var images = [
7. { "big":"http://i-2.yxdown.com/2015/3/18/KDkwMHgp/6381ccc0-ed65-4422-8671-b3158d6ad23e.jpg",
8. "thumb":"http://i-2.yxdown.com/2015/3/18/KHgxMjAp/6381ccc0-ed65-4422-8671-b3158d6ad23e.jpg",
9. "original":"http://i-2.yxdown.com/2015/3/18/6381ccc0-ed65-4422-8671-b3158d6ad23e.jpg",
10. "title":"","descript":"","id":75109},
11. { "big":"http://i-2.yxdown.com/2015/3/18/KDkwMHgp/fec26de9-8727-424a-b272-f2827669a320.jpg",
12. "thumb":"http://i-2.yxdown.com/2015/3/18/KHgxMjAp/fec26de9-8727-424a-b272-f2827669a320.jpg",
13. "original":"http://i-2.yxdown.com/2015/3/18/fec26de9-8727-424a-b272-f2827669a320.jpg",
14. "title":"","descript":"","id":75110},
15. </script>
16. '''
18. html\_script = r'<script>(.\*?)</script>'
19. m\_script = re.findall(html\_script,content,re.S|re.M)
20. for script in m\_script:
21. res\_original = r'"original":"(.\*?)"' *#原图*
22. m\_original = re.findall(res\_original,script)
23. for pic\_url in m\_original:
24. print pic\_url
25. filename = os.path.basename(pic\_url) *#去掉目录路径,返回文件名*
26. urllib.urlretrieve(pic\_url, 'E:\\'+filename) *#下载图片*

运行结果如下图所示，同时下载图片至E盘。  
参考文章： [[python学习] 简单爬取图片网站图库中图片](http://blog.csdn.net/eastmount/article/details/44492787)

------------------------------------------------------------------------------------------------------------------------------  
  
**9.通过replace过滤<br />标签**  
  
在获取值过程中，通常会存<br />标签，它表示HTML换行的意思。常用的方法可以通过标签'<'和'>'进行过滤，但是这里我想讲述的是一种Python常用的过滤方法，在处理中文乱码或一些特殊字符时，可以使用函数replace过滤掉这些字符。核心代码如下：  
**if '<br />' in value:  
        value = value.replace('<br />','')   #过滤该标签  
        value = value.replace('\n',' ')         #换行空格替代 否则总换行**  
例如过滤前后的例子：  
達洪阿 異名： (字) 厚菴<br /> (諡) 武壯<br /> (勇號) 阿克達春巴圖魯  
達洪阿 異名： (字) 厚菴 (諡) 武壯 (勇號) 阿克達春巴圖魯  
  
  
------------------------------------------------------------------------------------------------------------------------------  
  
**10.获取<img ../>中超链接及过滤<img>标签**  
  
在获取值属性值过程中，可能在分析table/tr/th/td标签后，仍然存在<img />图片链接，此时在获取文字内容时，你可能需要过滤掉这些<img>标签。这里采用的方法如下：  
        **value = re.sub('<[^>]+>','', value)**  
例如：

1. *#encoding:utf-8*
2. import os
3. import re
5. value = '''
6. <table class="infobox" style="width: 21em; text-align: left;" cellpadding="3">
7. <tr bgcolor="#CDDBE8">
8. <th colspan="2">
9. <center class="role"><b>中華民國政治人士</b><br /></center>
10. </th>
11. </tr>
12. <tr>
13. <th>性別：</th>
14. <td>男</td>
15. </tr>
16. <tr>
17. <th>政黨：</th>
18. <td><span class="org">
19. <img alt="中國國民黨" src="../../../../images/Kuomintang.svg.png" width="19" height="19" border="0" />
20. <a href="../../../../articles/%8B%E6%B0%91%E9%BB%A8.html" title="中國國民黨">中國國民黨</a></span></td>
21. </tr>
22. </table>
23. '''
25. value = re.sub('<[^>]+>','', value) *#过滤HTML标签*
26. print value

输出如下：

1. >>>
3. 中華民國政治人士
5. 性別：
6. 男
8. 政黨：
10. 中國國民黨
12. >>>

虽然仅仅包括汉字，但是中间会存在换行，需要过滤<br />即可：

1. if '<br />' in value:
2. value = value.replace('<br />','')
3. value = value.replace('\n',' ')
4. value = re.sub('<[^>]+>','', value) *#过滤HTML标签*
5. print value
7. *#输出仅仅一行如下：*
8. *#中華民國政治人士    性別： 男   政黨：   中國國民黨*

下面讲述第二部分，通过正则表达式获取<img>中的src超链接，代码如下：

1. test = '''<img alt="中國國民黨" src="../images/Kuomintang.png" width="19" height="19" border="0" />'''
2. print re.findall('src="(.\*?)"',test)

输出如下所示：

1. >>>
2. ['../images/Kuomintang.png']
3. >>>

findall函数返回的总是正则表达式在字符串中所有匹配结果的列表，即findall中返回列表中每个元素包含的信息。