pip install beautifulsoup4

pip install xlsxwriter

pip install Workbook

pip install itchat pillow

pip install pymongo

pip install pdfminer3k

pip install pytest-html

//pytest-HTML是一个插件，pytest用于生成测试结果的HTML报告。兼容Python 2.7,3.6

python学习之字符串前'r'的用法

在打开文件的时候open(r'c:\....')

加r和不加''r是有区别的:

1、'r'是防止字符转义的 如果路径中出现'\t'的话 不加r的话\t就会被转义 而加了'r'之后'\t'就能保留原有的样子

2、在字符串赋值的时候 前面加'r'可以防止字符串在时候的时候不被转义 原理是在转义字符前加'\'

例：

s=r'\tt' print(s) Output:'\tt'

s='\tt' print(s) Output:' t'

from bs4 import BeautifulSoup #1、导入库

headers={'User-Agent': 'Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.110 Safari/537.36','referer':"www.mmjpg.com" } #2、请求头herders

all\_url = 'http://www.mmjpg.com/'

'User-Agent':请求方式

'referer':从哪个链接跳转进来的

start\_html = requests.get(all\_url, headers=headers) #3、建立连接

all\_url：起始的地址，也就是访问的第一个页面；headers：请求头，告诉服务器是谁来了；requests.get：一个方法能获取all\_url的页面内容并且返回内容。

Soup = BeautifulSoup(start\_html.text, 'lxml') #4、解析获取的页面

BeautifulSoup：解析页面；lxml：解析器；start\_html.text：页面的内容

all\_a = Soup.find('div', class\_='pic').find\_all('a')[-2] #5、处理获取的页面

Soup.find（）查找某一个

find\_all（）查找所有的，返回一个列表

.find('img')['src'] ：获取img的src链接属性

class\_\_:获取目标的类名

div/a:类型条件为div/a的

[-2]可以用来去掉最后多匹配的标签，这里表示去掉最后两个a标签

<a href =# >内容</a> #6、获取目标内容

a[i]/get\_text():获取第i个a标签里面的内容

os.makedirs(os.path.join("E:\name", filename)) #7、可能用到的其他功能介绍：1、文件夹创建与切换

#在目录E:\name下创建名为filename的文件夹

os.chdir("E:\name\\" + filename)

#切换工作路径到E:\name\filename下

f = open(name+'.jpg', 'ab')##写入多媒体文件必须要 b 这个参数！#2、文件保存

f.write(img.content) ##多媒体文件要是用conctent！

f.close()

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)

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>' #获取<a href></a>之间的内容    
        mm =  re.findall(res, content, re.S|re.M)

#获取所有<a href></a>链接所有内容

urls=re.findall(r"<a.\*?href=.\*?<\/a>", content, re.I|re.S|re.M)

res\_url = r"(?<=href=\").+?(?=\")|(?<=href=\').+?(?=\')" #获取<a href></a>中的URL

或者res\_url=r'href="(.\*?)"'

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" \t "http://blog.csdn.net/Eastmount/article/details/_blank)

<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" \t "http://blog.csdn.net/Eastmount/article/details/_blank)  
此时需要通过该URL的"/"后面的参数命名图片，则方法如下：

urls = "http://i1.hoopchina.com.cn/blogfile/201411/11/BbsImg141568417848931\_640\*640.jpg"

values = urls.split('/')[-1]

print values

输出：BbsImg141568417848931\_640\*640.jpg

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：

url = 'http://localhost/test.py?a=hello&b=world'

values = url.split('?')[-1]

print values

for key\_value in values.split('&'):

    print key\_value.split('=')

输出如下所示：

a=hello&b=world

['a', 'hello']

['b', 'world']