

[python爬虫] 正则表达式使用技巧及爬取个人博客实例

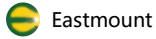
原创 Eastmount 最后发布于2017-10-18 18:15:59 阅读数 18873 ☆ 收藏

编辑 展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



¥9.90

去订阅

这篇博客是自己《数据挖掘与分析》课程讲到正则表达式爬虫的相关内容，主要简单介绍Python正则表达式爬虫，同时讲述常见的正则表达式分析方法，最后通过实例爬取作者的个人博客网站。希望这篇基础文章对您有所帮助，如果文章中存在错误或不足之处，还请海涵。真的太忙了，太长时间没有写博客了，抱歉~

一.正则表达式

正则表达式 (Regular Expression, 简称Regex或RE) 又称为正规表示法或常规表示法, 常常用来检索、替换那些符合某个模式的文本, 它首先设定好了一些特殊的字及字符组合, 通过组合的“规则字符串”来对表达式进行过滤, 从而获取或匹配我们想要的特定内容。它具有灵活、逻辑性和功能性非常的强, 能迅速地通过表达式从字符串中找到所需信息的优点, 但对于刚接触的人来说, 比较晦涩难懂。

1.re模块

Python通过re模块提供对正则表达式的支持, 使用正则表达式之前需要导入该库。

```
import re
```

其基本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一个匹配 (Match) 实例, 再使用Match实例获得所需信息。常用的函数是findall, 原型如下:

```
findall(string[, pos[, endpos]]) | re.findall(pattern, string[, flags])
```

该函数表示搜索字符串string, 以列表形式返回全部能匹配的子串。

其中参数re包括三个常见值:

- (1)re.I(re.IGNORECASE): 忽略大小写 (括号内是完整写法)
- (2)re.M(re.MULTILINE): 允许多行模式
- (3)re.S(re.DOTALL): 支持点任意匹配模式

Pattern对象是一个编译好的正则表达式, 通过Pattern提供的一系列方法可以对文本进行匹配查找。Pattern不能直接实例化, 必须使用re.compile()进行构造。

2.compile方法

re正则表达式模块包括一些常用的操作函数, 比如compile()函数。其原型如下:

```
compile(pattern[, flags] )
```

该函数根据包含正则表达式的字符串创建模式对象, 返回一个pattern对象。参数flags是匹配模式, 可以使用按位或 “|”

表示同时生效，也可以在正则表达式字符串中指定。Pattern对象是不能直接实例化的，只能通过compile方法得到。

简单举个实例，使用正则表达式获取字符串中的数字内容，如下所示：

```
>>> import re
>>> string="A1.45, b5, 6.45, 8.82"
>>> regex = re.compile(r"\d+\.\d*")
>>> print regex.findall(string)
['1.45', '5', '6.45', '8.82']
>>>
```

3.match方法

match方法是从字符串的pos下标处起开始匹配pattern，如果pattern结束时已经匹配，则返回一个Match对象；如果匹配过程中pattern无法匹配，或者匹配未结束就已到达endpos，则返回None。该方法原型如下：

```
match(string[, pos[, endpos]]) | re.match(pattern, string[, flags])
```

参数string表示字符串；pos表示下标，pos和endpos的默认值分别为0和len(string)；参数flags用于编译pattern时指定匹配模式。

4.search方法

search方法用于查找字符串中可以匹配成功的子串。从字符串的pos下标处起尝试匹配pattern，如果pattern结束时仍可匹配，则返回一个Match对象；若无法匹配，则将pos加1后重新尝试匹配；直到pos=endpos时仍无法匹配则返回None。函数原型如下：

```
search(string[, pos[, endpos]]) | re.search(pattern, string[, flags])
```

参数string表示字符串；pos表示下标，pos和endpos的默认值分别为0和len(string)；参数flags用于编译pattern时指定匹配模式。

5.group和groups方法

group([group1, ...])方法用于获得一个或多个分组截获的字符串，当它指定多个参数时将以元组形式返回。

groups([default])方法以元组形式返回全部分组截获的字符串，相当于调用group(1,2,...last)。default表示没有截获字符串的组以这个值替代，默认为None。

二.正则表达式抓取网络数据常见方法

在第三小节作者将介绍常用的正则表达式抓取网络数据的一些技巧，这些技巧都是作者自然语言处理和数据抓取实际编程中的总结，可能不是很系统，但是也能给读者提供一些抓取数据的思路以及解决实际的一些问题。

1.抓取标签间的内容

HTML语言是采用标签对的形式来编写网站的，包括起始标签和结束标签，比如<head></head>、<tr></tr>、

<script> <script>等。下面讲解抓取标签对之间的文本内容。

(1) 抓取title标签间的内容

首先爬取网页的标题，采用的正则表达式为'<title>(.*?)</title>', 爬取百度标题代码如下：

```
# coding=utf-8
import re
import urllib
url = "http://www.baidu.com/"
content = urllib.urlopen(url).read()
title = re.findall(r'<title>(.*?)</title>', content)
print title[0]
# 百度一下，你就知道
```

代码调用urllib库的urlopen()函数打开超链接，并借用正则表达式库中的findall()函数寻找title标签间的内容，由于findall()函数获取所有满足该正则表达式的文本，故输出第一个值title[0]即可。下面是获取标签的另一种方法。

```
pat = r'(<=title>).*?(?=</title>)'
ex = re.compile(pat, re.M|re.S)
obj = re.search(ex, content)
title = obj.group()
print title
# 百度一下，你就知道
```

(2) 抓取超链接标签间的内容

在HTML中， 用于标识超链接，test03_08.py文件用于获取完整的超链接和超链接<a>和之间的内容。

```
# coding=utf-8
import re
import urllib
url = "http://www.baidu.com/"
content = urllib.urlopen(url).read()

# 获取完整超链接
res = r"<a.*?href=.*?</a>"
urls = re.findall(res, content)
for u in urls:
    print unicode(u, 'utf-8')

# 获取超链接<a>和</a>之间内容
res = r'<a .*?>(.*?)</a>'
texts = re.findall(res, content, re.S|re.M)
for t in texts:
    print unicode(t, 'utf-8')
```

输出结果部分内容如下所示，这里如果直接输出print u或print t可能会乱码，需要调用函数unicode(u,'utf-8')进行转码。

```
# 获取完整超链接
<a href="http://news.baidu.com" name="tj_trnews" class="mnav">新闻</a>
<a href="http://www.hao123.com" name="tj_trhao123" class="mnav">hao123</a>
<a href="http://map.baidu.com" name="tj_trmap" class="mnav">地图</a>
<a href="http://v.baidu.com" name="tj_trvideo" class="mnav">视频</a>
```

```
...  
  
#获取超链接<a>和</a>之间内容  
新闻  
hao123  
地图  
视频  
...
```

(3) 抓取tr\td标签间的内容

网页中常用的布局包括table布局或div布局，其中table表格布局中常见的标签包括tr、th和td，表格行为tr（table row），表格数据为td（table data），表格表头th（table heading）。那么如何抓取这些标签之间的内容呢？下面代码是获取它们之间内容。

假设存在HTML代码如下所示：

```
<html>  
<head><title>表格</title></head>  
<body>  
    <table border=1>  
        <tr><th>学号</th><th>姓名</th></tr>  
        <tr><td>1001</td><td>杨秀璋</td></tr>  
        <tr><td>1002</td><td>严娜</td></tr>  
    </table>  
</body>  
</html>
```

则爬取对应值的Python代码如下：

```
# coding=utf-8  
import re  
import urllib  
content = urllib.urlopen("test.html").read() #打开本地文件  
  
#获取<tr></tr>间内容  
res = r'<tr>(.*?)</tr>'  
texts = re.findall(res, content, re.S|re.M)  
for m in texts:  
    print m  
  
#获取<th></th>间内容  
for m in texts:  
    res_th = r'<th>(.*?)</th>'  
    m_th = re.findall(res_th, m, re.S|re.M)  
    for t in m_th:  
        print t  
  
#直接获取<td></td>间内容  
res = r'<td>(.*?)</td><td>(.*?)</td>'  
texts = re.findall(res, content, re.S|re.M)  
for m in texts:  
    print m[0],m[1]
```

输出结果如下，首先获取tr之间的内容，然后再在tr之间内容中获取<th>和</th>之间值，即“学号”、“姓名”，最后讲述直接获取两个<td>之间的内容方法。

```
>>>
<th>学号</th><th>姓名</th>
<td>1001</td><td>杨秀璋</td>
<td>1002</td><td>严娜</td>

学号
姓名

1001 杨秀璋
1002 严娜
>>>
```

2. 抓取标签中的参数

(1) 抓取超链接标签的URL

HTML超链接的基本格式为 “链接内容”，现在需要获取其中的URL链接地址，方法如下：

```
# coding=utf-8
import re

content = '''
<a href="http://news.baidu.com" name="tj_trnews" class="mnav">新闻</a>
<a href="http://www.hao123.com" name="tj_trhao123" class="mnav">hao123</a>
<a href="http://map.baidu.com" name="tj_trmap" class="mnav">地图</a>
<a href="http://v.baidu.com" name="tj_trvideo" class="mnav">视频</a>
'''

res = r"(?<=href=\"").+?(?=\")|(?<=href=\'\').+?(?=\')"
urls = re.findall(res, content, re.I|re.S|re.M)
for url in urls:
    print url
```

输出内容如下：

```
>>>
http://news.baidu.com
http://www.hao123.com
http://map.baidu.com
http://v.baidu.com
>>>
```

(2) 抓取图片超链接标签的URL

HTML插入图片使用标签的基本格式为 “”，则需要获取图片URL链接地址的方法如下：

```
content = ''''''
urls = re.findall('src="(.*?)"', content, re.I|re.S|re.M)
print urls
# ['http://www..csdn.net/eastmount.jpg']
```

其中图片对应的超链接为 “http://www..csdn.net/eastmount.jpg”，这些资源通常存储在服务器端，最后一个 “/” 后面的字段即为资源的名称，该图片名称为 “eastmount.jpg”。那么如何获取URL中最后一个参数呢？

(3) 获取URL中最后一个参数

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，如前面的“eastmount.jpg”，需要通过URL “/” 后面的参数获取图片。

```
content = ''''''
urls = 'http://www..csdn.net/eastmount.jpg'
name = urls.split('/')[ -1]
print name
# eastmount.jpg
```

该段代码表示采用字符 “/” 分割字符串，并且获取最后一个获取的值，即为图片名称。

3.字符串处理及替换

在使用正则表达式爬取网页文本时，通常需要调用find()函数找到指定的位置，再进行进一步爬取，比如获取class属性为“infobox”的表格table，再进行定位爬取。

```
start = content.find(r'<table class="infobox"') #起点位置
end = content.find(r'</table>') #重点位置
infobox = text[start:end]
print infobox
```

同时爬取过程中可能会爬取到无关变量，此时需要对无关内容进行过滤，这里推荐使用replace函数和正则表达式进行处理。比如，爬取内容如下：

```
# coding=utf-8
import re

content = '''
<tr><td>1001</td><td>杨秀璋<br /></td></tr>
<tr><td>1002</td><td>颜 娜</td></tr>
<tr><td>1003</td><td><B>Python</B></td></tr>
'''

res = r'<td>(.*?)</td><td>(.*?)</td>'
texts = re.findall(res, content, re.S|re.M)
for m in texts:
    print m[0],m[1]
```

输出如下所示：

```
>>>
1001 杨秀璋<br />
1002 颜 娜
1003 <B>Python</B>
>>>
```

此时需要过滤多余字符串，如换行（
）、空格（ ）、加粗（）。过滤代码如下：

```
# coding=utf-8 import re

content = '''
<tr><td>1001</td><td>杨秀璋<br /></td></tr>
<tr><td>1002</td><td>颜 娜</td></tr>
<tr><td>1003</td><td><B>Python</B></td></tr>
'''

res = r'<td>(.*?)</td><td>(.*?)</td>'
texts = re.findall(res, content, re.S|re.M)
for m in texts:
    value0 = m[0].replace('<br />', '').replace(' ', '')
    value1 = m[1].replace('<br />', '').replace(' ', '')
    if '<B>' in value1:
        m_value = re.findall(r'<B>(.*?)</B>', value1, re.S|re.M)
        print value0, m_value[0]
    else:
        print value0, value1
```

采用replace将字符串“
”或“ ”替换成空白，实现过滤，而加粗（）需要使用正则表达式过滤，输出结果如下：

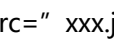
```
1 >>>
2 1001 杨秀璋
3 1002 颜娜
4 1003 Python
5 >>>
```

三.实战爬取个人博客实例

在讲述了正则表达式、常用网络数据爬取模块、正则表达式爬取数据常见方法等内容之后，我们将讲述一个简单的正则表达式爬取网站的实例。这里作者用正则表达式爬取作者的个人博客网站的简单示例，获取所需内容。作者的个人网址“http://www.eastmountyxz.com/”打开如下图所示。



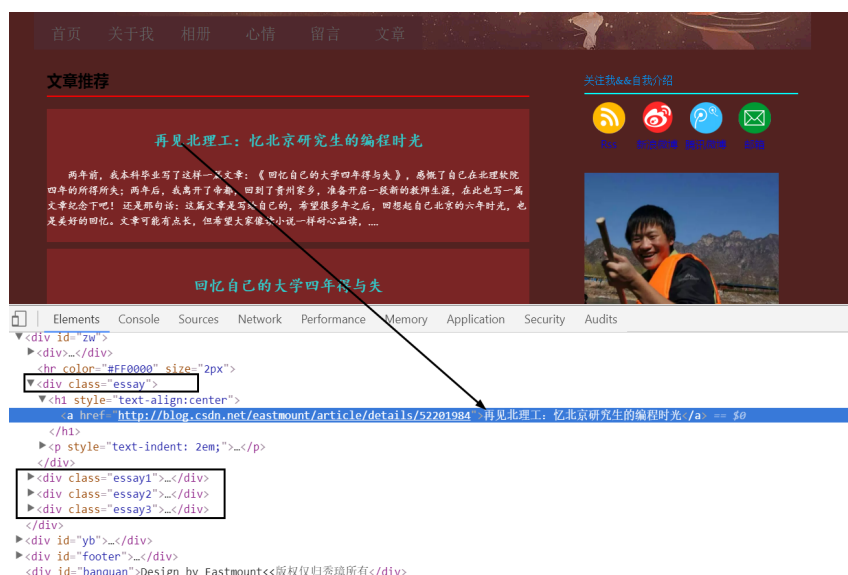
假设现在需要爬取的内容如下：

1. 博客网址的标题（title）内容。
2. 爬取所有图片的超链接，比如爬取中的“xxx.jpg”。
3. 分别爬取博客首页中的四篇文章的标题、超链接及摘要内容，比如标题为“再见北理工：忆北京研究生的编程时光”。

1. 分析过程

第一步 浏览器源码定位

首先通过浏览器定位这些元素源代码，发现它们之间的规律，这称为DOM树文档节点树分析，找到所需爬取节点对应的属性和属性值，如图3.6所示。



标题“再见北理工：忆北京研究生的编程时光”位于<div class=“ essay” ></div>节点下，它包括一个<h1></h1>记录标题，一个<p></p>记录摘要信息，即：

```
1 | <div class="essay">
2 | <h1 style="text-align:center">
3 | <a href="http://blog.csdn.net/eastmount/.../52201984">
4 | 再见北理工：忆北京研究生的编程时光
5 | </a>
6 | </h1>
7 | <p style="text-indent: 2em;">
8 |
```

两年前，我本科毕业写了这样一篇文章：《回忆自己的大学四年得与失》，感慨了自己在北理软院四年的所得所失；两年后，我离开了帝都，回到了贵州家乡，准备开启一段新的教师生涯，在此也写一篇文章纪念下吧！

```
9 |
10 | 还是那句话：这篇文章是写给自己的，希望很多年之后，回想起自己北京的六年时光，也是美好的回忆。文章可能有点长，但希望大家像读小说一样
    | 耐心品读，....
```

```
10 | </p>11 | </div>
```

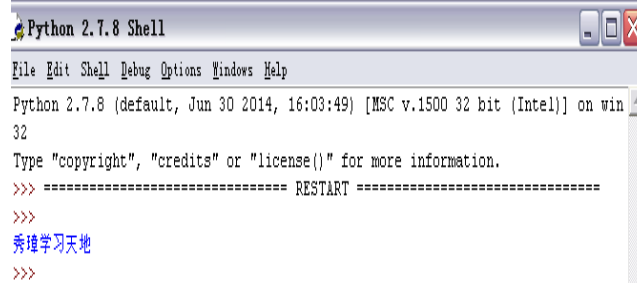
其余三篇文章同样为<div class=“ essay1” ></div>、<div class=“ essay2” ></div>和<div class=“ essay3” ></div>。

第二步 正则表达式爬取标题

网站的标题通常位于<head><title>...</title></head>之间，爬取博客网站的标题“秀璋学习天地”的方法是通过正则表达式“<title>(.*?)</title>”实现，代码如下，首先通过urlopen()函数访问博客网址，然后定义正则表达式爬取。如下图所示：

```
import re
import urllib

url = "http://www.eastmountyxz.com/"
content = urllib.urlopen(url).read()
title = re.findall(r'<title>(.*?)</title>', content)
print title[0]
```



Python 2.7.8 Shell

File Edit Shell Debug Options Windows Help

Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win32

Type "copyright", "credits" or "license()" for more information.

>>> ===== RESTART =====

>>>

秀璋学习天地

>>>

第三步 正则表达式爬取所有图片地址

由于HTML插入图片标签格式为“”，则使用正则表达式获取图片URL链接地址的方法如下，获取以“src=”开头，以双引号结尾的内容即可。

```
1 import re
2 import urllib
3
4 url = "http://www.eastmountyxz.com/"
5 content = urllib.urlopen(url).read()
6 urls = re.findall(r'src="(.*?)"', content)
7 for url in urls:
8     print url
```

输出共显示了6张图片，但每张图片省略了博客地址“http://www.eastmountyxz.com/”，增加相关地址则可以通过浏览器访问，如“http://www.eastmountyxz.com/images/11.gif”。

第四步 正则表达式爬取博客内容

前面第一步讲述了如何定位四篇文章的标题，第一篇文章位于<div class=“ essay”>和</div>标签之间，第二篇位于<div class=“ essay1”>和</div>，依次类推。但是该HTML代码存在一个错误：class属性通常表示一类标签，它们的值都应该是相同的，所以这四篇文章的class属性都应该是“essay”，而name或id可以用来标识其唯一值。

这里使用find()函数定位<div class=“ essay”>开头，</div>结尾，获取它们之间的值。比如获取第一篇文章的标题和超链接代码如下：

```
1 import re
2 import urllib
3 url = "http://www.eastmountyxz.com/"
4 content = urllib.urlopen(url).read()
```

```

5 | start = content.find(r'<div class="essay">') 6 | end = content.find(r'<div class="essay1">')
7 | print content[start:end]

```

该部分代码分为三步骤：

- (1) 调用urllib库的urlopen()函数打开博客地址，并读取内容赋值给content变量。
- (2) 调用find()函数查找特定的内容，比如class属性为“essay”的div标签，依次定位获取开始和结束的位置。
- (3) 进行下一步分析，获取源码中的超链接和标题等内容。

定位这段内容之后，再通过正则表达式获取具体内容，代码如下：

```

1 | import re
2 | import urllib
3 |
4 | url = "http://www.eastmountyxz.com/"
5 | content = urllib.urlopen(url).read()
6 | start = content.find(r'<div class="essay">')
7 | end = content.find(r'<div class="essay1">')
8 | page = content[start:end]
9 |
10 | res = r"(?<=href=\").+?(?=\")|(?<=href=\').+?(?=\')\"
11 | t1 = re.findall(res, page) #超链接
12 | print t1[0]
13 | t2 = re.findall(r'<a .*?>(.*?)</a>', page) #标题
14 | print t2[0]
15 | t3 = re.findall('<p style=.*?>(.*?)</p>', page, re.M|re.S) #摘要(
16 | print t3[0]

```

调用正则表达式分别获取内容，由于爬取的段落（P）存在换行内容，所以需要加入re.M和re.S支持换行查找，最后输出结果如下：

```

1 | >>>
2 | http://blog.csdn.net/eastmount/article/details/52201984
3 | 再见北理工：忆北京研究生的编程时光
4 |
   | 两年前，我本科毕业写了这样一篇文章：《回忆自己的大学四年得与失》，感慨了自己在北理软院四年的所得所失；两年后，我离开了帝都，回
   | 到了贵州家乡，准备开启一段新的教师生涯，在此也写一篇文章纪念下吧！
5 | 6 |
   | 还是那句话：这篇文章是写给自己的，希望很多年之后，回想起自己北京的六年时光，也是美好的回忆。文章可能有点长，但希望大家像读
   | 小说一样耐心品读，....
7 | >>>

```

2.代码实现

完整代码参考test03_10.py文件，代码如下所示。

```

1 | #coding:utf-8
2 | import re
3 | import urllib
4 |
5 | url = "http://www.eastmountyxz.com/"
6 | content = urllib.urlopen(url).read()
7 |
8 | #爬取标题
9 | title = re.findall(r'<title>(.*?)</title>', content)
10 | print title[0]

```

```

11 | 12 | #爬取图片地址
13 | urls = re.findall(r'src="(.*?)"', content)
14 | for url in urls:
15 |     print url
16 |
17 | #爬取内容
18 | start = content.find(r'<div class="essay">')
19 | end = content.find(r'<div class="essay1">')
20 | page = content[start:end]
21 | res = r"(?<=href=\\").+?(?=\\")|(?<=href=\\').+?(?=\\'")
22 | t1 = re.findall(res, page) #超链接
23 | print t1[0]
24 | t2 = re.findall(r'<a .*?>(.*?)</a>', page) #标题
25 | print t2[0]
26 | t3 = re.findall('<p style=.*?>(.*?)</p>', page, re.M|re.S) #摘要(
27 | print t3[0]
28 | print ''
29 |
30 | start = content.find(r'<div class="essay1">')
31 | end = content.find(r'<div class="essay2">')
32 | page = content[start:end]
33 | res = r"(?<=href=\\").+?(?=\\")|(?<=href=\\').+?(?=\\'")
34 | t1 = re.findall(res, page) #超链接
35 | print t1[0]
36 | t2 = re.findall(r'<a .*?>(.*?)</a>', page) #标题
37 | print t2[0]
38 | t3 = re.findall('<p style=.*?>(.*?)</p>', page, re.M|re.S) #摘要(
39 | print t3[0]

```

输出结果如图所示。

```

>>>
秀璋学习天地
./images/11.gif
./images/04.gif
./images/05.gif
./images/06.gif
./images/07.gif
./images/08.jpg
http://blog.csdn.net/eastmount/article/details/52201984
再见北理工：忆北京研究生的编程时光
    两年前，我本科毕业写了这样一篇文章：《回忆自己的大学四年得与失》，感慨了自己在北理软院
    四年的所得所失；两年后，我离开了帝都，回到了贵州家乡，准备开启一段新的教师生涯，在此也写一篇
    文章纪念下吧！
    还是那句话：这篇文章是写给自己的，希望很多年之后，回想起自己北京的六年时光，也是美
    好的回忆。文章可能有点长，但希望大家像读小说一样耐心品读，....

http://blog.csdn.net/eastmount/article/details/34619941
回忆自己的大学四年得与失
转眼间，大学四年就过去了，我一直在犹豫到底要不要写一篇文章来回忆自己大学四年的所得所失，最后还
是准备写下这样一篇文章来纪念自己的大学四年吧！这篇文章是写给自己的，多少年之后回想起自己的大学
青春也是美好的回忆，也希望大家像读小说一样看完后也能温馨一笑或唏嘘摇头，想想自己的大学生活吧！
如果有错误或不足之处，请海涵或当做荒唐之言即可....
>>>

```

通过上面的代码，读者会发现使用正则表达式爬取网站还是比较繁琐，尤其是定位网页节点时，后面将讲述Python提供的常用第三方扩展包，利用这些包的函数进行定向爬取。

希望这篇文字对你有所帮助，尤其是刚接触爬虫的同学或是遇到类似问题的同学，更推荐大家使用BeautifulSoup、Selenium、Scrapy等库来爬取数据。

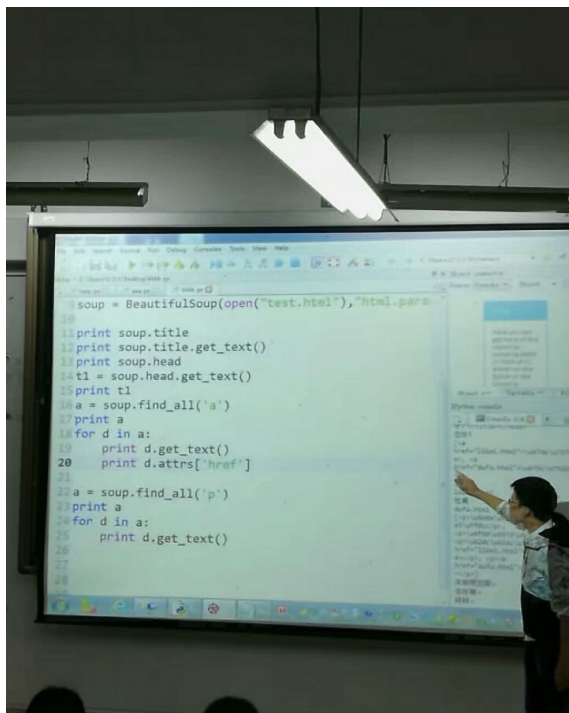
总结:

贵州纵美路迢迢,
未负劳心此一遭。
搜得破书三四本,
也堪将去教尔曹。

这是我大学毕业离开北京时写的诗，回想依然感慨万分，当时放弃了互联网月薪过万的工资，亲朋好友的劝阻，选择回贵州任教，子承父志。刚来财大领了两个月2800的工资，但内心始终都是愉悦的。

转眼已工作一年多，自己也有了新的感悟。知道了有一些事情比事业重要得多，即使是最喜欢的教书育人，也可以放弃，似乎却总让你操心，确实不该。人，一方面需要牢记自己的初心，能坚持做一辈子喜欢的事真的很难，所以教书的我是幸运的; 另一方面也要学会say no，那时的秀璋才真正成长。

人生得一知己足矣，教育和工作都是根植于爱中。最近对不住很多人，加班太多了，都是深夜一两点，中午和坐公交的短暂时间都用来学习了，但很多编程问题都还来不及解答，房子也没关心，博客也来不及撰写。唉，但想想她、亲人、朋友和学生，我接着擦干眼泪继续前行，还有有你。这就是生活吗？忙碌的你们也要注意休息哈。附一张上课图片，忙取了。



(By:Eastmount 2017-10-19 清早9点 <http://blog.csdn.net/eastmount/>)

👍 点赞 27 ☆ 收藏 🔄 分享



Eastmount 博客专家

发布了444 篇原创文章 · 获赞 5918 · 访问量 484万+