Python版本：2.7

首先抓取<http://m.sohu.com的html>

f=urllib2.urlopen('http://m.sohu.com')

buf=f.read()

print buf

运行看一下buf的显示的html代码

因为建立的文件夹格式为201506161322，已知路径'D:\tmp\backup\'

插入建立文件夹的模块os

写建立文件夹的函数，

def makedr(title,path):

new\_path = os.path.join(path, title)

if not os.path.isdir(new\_path):

os.makedirs(new\_path)

return new\_path

title是新建的文件夹名 类型str

path是建立文件夹的路径，我们要把sohu.html放在20150616。。。文件夹下，path为'D:\tmp\backup\'，title=’2015。。。’

现在我们需要根据系统时间得出文件夹的名字

这里我们可以用time模块

def filepath():

now = time.localtime()

year=str(now.tm\_year)

mon=now.tm\_mon

if mon<10:

mon=''.join(['0',str(mon)])

else:

mon=str(mon)

day=now.tm\_mday

if day<10:

day=''.join(['0',str(day)])

else:

day=str(day)

hour=now.tm\_hour

if hour<10:

hour=''.join(['0',str(hour)])

else:

hour=str(hour)

mmin=now.tm\_min

if mmin<10:

mmin=''.join(['0',str(mmin)])

else:

mmin=str(mmin)

dirname=''.join([year,mon,day,hour,mmin])

return dirname

但是由于我们的文件夹格式的原因，当月份天数小时分钟小于十的时候要在前面补零。加入判断语句，最后将字符串连接起来，返回文件夹的名字。

接下来就是程序的主体部分，正则表达式在buf里提取我们想要的文件存储路径。

对于.js.css.jpg.png等结尾的文件，我们的正则表达式书写比较简单。分别提取文件路径。如<http://s1.rr.itc.cn/h5/js/tags/v3/msohu/3.1.29/home.js>根据这个路径我们要在本地文件夹保存相对应的文件。由于这个地方图片，js，css操作都是一样的。我就以js做例子。这里我们要在对应的文件夹下建立js文件夹。所以我们还要调用建立文件夹的函数。Path前面已经得到了，title=‘js’。我们又可以得到存储js文件夹的路径。假设是jsPath。下面给出存储文件的函数：addr我们传入的即为我们上面正则表达式提取出来的文件路径，path我们传入jsPath。在这里我们用split()将文件名提取和pop（）弹出文件名。在jsPath存入js文件。

def getFile(addr,path):

u = urllib2.urlopen(addr)

data = u.read()

splitPath = addr.split('/')

fName = splitPath.pop()

f = open('%s\\%s'%(path,fName), 'wb')

f.write(data)

f.close()

最后加上循环程序，我们还是用time模块判断条件为stopTime-startTime>=60:（=60有可能正好取不到）在运行程序的时候发现开始运行的会立刻循环两次，但是因为后面的会覆盖前面的，所以不影响，之后程序正常运行。

再然后查找buf的时候，发现还有一些内部的样式表，js没有被检测到。所以我们要根据<script>和<style>再次书写正则表达式。这个地方由于我们要找的内部js和css中间是有换行符的。查看了re模块，找到与多行字符串匹配的函数re.findall(‘’, ,re.S)，查看html关于脚本的标准时，我们发现js都是由<script type="text/javascript">开头，以</script>结尾。这样我们可以列出一个正则表达式匹配addrjs=re.findall('\<script.\*?\<\/script>',buf,re.S)，（为了简单，详细的可以将首尾都写上。）这里我们要将内部的js和外部的js分开，外部的js由于有网页路径，可以直接保存。我们用pop（）每次弹出list的最后一个，直到所有的list的元素全部取出。在弹出其中一个元素的时候，我们用singleJs1=re.findall('http.\*?\.js',singleJs)判断是否是内部js。外部的js的存储上面已经说过。内部的js我们要保存的是<script type="text/javascript"> ，</script>中间部分。我们用singleJs=singleJs.split('</script>')，singleJs=singleJs.split('<script type="text/javascript">')将头尾去掉，并且pop（）留下来的空字符串。最后得到中间的代码，保存在js文件夹下，这里我为了简单用数字直接命名。命名为1-n.js。上一部分的js存储这里我们直接抛弃不用，但是css内部样式表格式与外部样式表格式不同，所以css还是要加上上一部分基于已知网页路径存储css文件那一部分。