链滴

爬虫刷题

作者：kanadeblisst
原文链接：https://ld246.com/article/1565948671496
来源网站：链滴

今天偶然得到一本爬虫秘籍，现在和大家交流分享一下思路。(网站作者做的很好，支持一下)</p

网站链接：http://glidedsky.com/
<h2 id="第一题">第一题</h2>
这里有一个网站，里面有一些数字。把这些数字的总和，输入到答案框里面，即可通过本关。<b>

思路：这个很基础，只需要 requests 请求一下，再把数字提出来相加即可
<h2 id="第二题">第二题</h2>
爬虫往往不能在一个页面里面获取全部想要的数据，需要访问大量的网页才能够完成任务。这里一个网站，还是求所有数字的和，只是这次分了 1000 页。

思路：只需要构造链接迭代请求一下即可
<h2 id="第三题-IP屏蔽-">第三题（IP 屏蔽）</h2>
你的每个 IP，只能访问一次，之后就会被封禁。悄悄地告诉你，你之前用过的 IP，已经被悄记录了~这里有一个网站，分了 1000 页，求所有数字的和。
思路：换代理就行，不过 1000 个代理我可没有，下一题
<h2 id="第四题--字体反爬-">第四题 （字体反爬）</h2>
如果字符 0 展示并不是 0 的图像是 1 的图像呢？这也就意味着爬虫拿到的是字符 0，但是人看的却是图像 1。而我们知道，一切从字符到图像的映射，都可以用来反爬。这有一个网站，分了 1000 页，求所有数字的和。注意，是人看到的数字，不是网页源码中的数字哦~<r>

思路：一开始想到的肯定是直接把字符的对应关系找出来，然后直接替换即可。但是这仅仅满足只有一套字体文件，但这个网站每个页面都有一个字体文件。这样的话，解决方法只有两种了：解析体文件、OCR 识别。OCR 的话其实很少在爬虫应用，因为效率实在是低，只有破解验证码这种工作少的工作才有涉及。所以我们还是破解字体文件吧。
步骤：

这个就是网站自己定义的字体文件，如果没有看过正常的字体文件，你可能不知道有什么区别。我们一下正常的字体文件：

那么该如何处理这种文件呢，我们可以使用 fonttools 这个 Python 库来将字体文件转化为 xml 式的，这样就可以用文本格式打开。转化代码：
<pre><code class="highlight-chroma">from fontTools.ttLib import TTFont

font = TTFont('a.off')
font.saveXML('a.xl')

</code></pre>

我们可以下载两套字体，都转为 xml 格式，然后使用文本比较工具（wincmp）来比较两套字体
直接看结果：

先简单介绍一下 xml 格式的字体格式，每个文字都有 ID 和对应的 name，而 name 也有其对应 ode，在最后会根据 name 来绘制文字（文字的一笔一划），而自定义的字体要么更改了 name 和 c de 的对应关系，比如明明 name 绘制的字体是 1，而他对应的 code 编码确实 0x30（0 的编码）。有一些更加奇葩，他直接将 name 绘制的一笔一画都重新构造。这样就要比对字体的相似度，这是一非常耗 CPU 的工作，一般不会去做。我们还是接着看这个网站的字体吧，经过比对发现，只有 code 和 name 对应关系改变了。后面的绘制代码一点没变。这样就容易解决了。先说一下这种的解决思路已知 name 对应的文字不变（手工找出 name 对应的文字是哪些），那么我们只需找到 code 和 na e 的对应关系就知道那个文字被更换成那个文字了。找到这样一个对应关系的字典，然后提取网页文的时候做替换就行。
<h2 id="爬虫-雪碧图1">爬虫-雪碧图 1</h2>
HTTP 是基于 TCP 连接的，TCP 连接的建立是需要时间和资源的。而下载网页所需的图片资源通过 HTTP 的。如果有非常多的小图片，就需要建立很多 TCP 连接。勤劳勇敢的前端工作者们，想把所有小图片放到一张图片里面去。这样就可以通过一次 TCP 链接，下载所有的小图片，再通过前的奇技淫巧，来展示正确的图片。这种由很多小图片组成的图片，被称为雪碧图。雪碧图可以节约 TC 连接的同时，也为爬取带来了难度。这里有一个网站，分了 1000 页，求所有数字的和。
思路：如果你没有见过这个，你肯定看不懂上面这些文字的含义。我们看看图：

sprite 这个 css 属性定义了一张背景图片，而前面那个属性则定义了一个偏移量，通过偏移量来制背景图片的位置来显示背景图片的某个文字。这就是这里所称的雪碧图。解决思路：图片是 base64 格式的，转化成图片是这样的。

通过踩点可以知道，每个网页都有 10 个偏移量，个个网页的偏移量会有些许变化，开始的思路我们访问每个网页的时候，提取所有偏移量然后排序，再顺序提取 css 属性和其偏移量。找出偏移量排序后的位置就可以知道偏移量和属性对应的是哪个文字。后面测试发现，有些网页的偏移量少于 10 个，应该是这个网页的文字有些没有。这就导致排序无法解决了。那么我们再完善一下，我们随便选一对偏移量做样本，然后将网页和样本顺序比较，如果发现差值大于 6，则认为网页的这个文字不存 。我们则再样本这个位置插入样本对应的值。这样迭代次数为 10-网页偏移量去重后的个数。这样就以把 css 属性对应的偏移量转化为网页显示的汉字。不过运行之后发现结果不对，我挑出几个结果和页比对，并没有什么错误。这原因就不得而知。
目前就做了这几道题，其他的以后在更新吧。