**爬虫进阶开发——xpath选择器常见用法**

俗话说，工欲上其事，必先利其器，学好xpath选择器，能极高的提升在爬虫的数据提取环节中的提取速度，下面我们来认识认识xpath。

**选取节点**

XPath 使用路径表达式在 XML 文档中选取节点。节点是通过沿着路径或者 step 来选取的。

**下面列出了最有用的路径表达式**

| **表达式** | **描述** |
| --- | --- |
| nodename | 选取此节点的所有子节点。 |
| / | 从根节点选取。 |
| // | 从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置。 |
| . | 选取当前节点。 |
| .. | 选取当前节点的父节点。 |
| @ | 选取属性。 |

**实例**

**1、精确查询**

$html =<<<STR

<div id="demo">

<span class="tt">bbb</span>

<span>ccc</span>

<p rel="pnode">ddd</p>

</div>

STR;

// 获取id为demo的div内容

$data = selector::select($html, "//div[@id='demo']");

// 获取class为tt的span内容

$data = selector::select($html, "//div[@class='tt']");

// 获取rel为pnode的p内容

$data = selector::select($html, "//div[@rel='pnode']");

**2、模糊查询**

contains 匹配一个属性值中包含的字符串

$html =<<<STR

<div id="demo1">

demo1

</div>

<div id="demo2">

demo2

</div>

STR;

// 查找id属性中包含demo关键字的页面元素

// 这里能获取id为demo1和demo2的内容

$data = selector::select($html, "//div[contains(@id,'demo')]");

**3、获取节点属性**

$html =<<<STR

<td data-value="3.80">3.80</td>

<td data-value="3.80">3.80</td>

<td data-value="3.80">3.80</td>

<td data-value="3.80">3.80</td>

STR;

// 获取 td 的 data-value 属性

$data = selector::select($html, "//td@data-value");

**XPATH的几个常用函数**

1.contains ()： //div[contains(@id, 'in')] ,表示选择id中包含有’in’的div节点

2.text()：由于一个节点的文本值不属于属性，比如<a class=”baidu“ href=”http://www.baidu.com“>baidu</a>,所以，用text()函数来匹配节点：//a[text()='baidu']

3.last()：//div[contains(@id, 'in')][las()]，表示选择id中包含有'in'的div节点的最后一个节点

4.starts-with()： //div[starts-with(@id, 'in')] ，表示选择以’in’开头的id属性的div节点

5.not()函数，表示否定，//input[@name=‘identity’ and not(contains(@class,‘a’))] ，表示匹配出name为identity并且class的值中不包含a的input节点。 not()函数通常与返回值为true or false的函数组合起来用，比如contains(),starts-with()等，但有一种特别情况请注意一下：我们要匹配出input节点含有id属性的，写法如下：//input[@id]，如果我们要匹配出input节点不含用id属性的，则为：//input[not(@id)]