## 一、Xpath 预览

官方文档为：https://www.w3.org/TR/xpath/

* 路径搜索语言 XML path language

在做爬虫时，使用xpath来做相应的信息抽取，提供了简介明了的路径选择表达式

常用规则有：

node name 选择此节点的所有子节点

/ 从当前节点选取直接子节点

// 从当前节点选取子孙节点

. 选取当前节点

.. 选取当前节点的父节点

@ 选取属性

//title[@lang=’en’]

* 选取当前的所有名称为title，属性为lang的值为en的节点

### 1、实例引入

到此，请阅读

##### 4.1.4.1.py和4.1.4.2.py

### 2、所有结点

**我们用//开头的XPath规则来选取所有符合要求的节点**

**//\*匹配所有节点**

##### 4.1.4.3.py

### 3、子节点

##### 4.1.4.4.py

**//li/a : 用于获取所有li节点的所有直接a节点**

**//ul//a : 获取ul节点下的所有子孙a节点**

**//ul/a : 没有任何输出，因为ul下没有直接的a子节点**

#### 区别：

|  |  |
| --- | --- |
| / | 用于获取直接子节点 |
| // | 用于获取子孙节点 |

### 4、父节点

##### 4.1.4.5.py

**..**  ：用于查找父节点

**parent:: 获取父节点**

### 5、属性匹配

选取的时候，可以使用@符号来进行属性匹配：

##### 4.1.4.6.py

### 6、文本获取

Xpath的text()方法获取节点的文本，注意是text()，而不是text

如果想要获取子孙节点内部的所有文本，可以直接用//加text()的方式，这样可以保证获取到最全面的文本信息，但是可能会夹杂一些换行符等特殊符号。

如果想要获取某些特定子孙节点下的所有文本，可以先选取到特定的子孙节点，然后再调用text()方法获取其内部文本，这样，可以保证获取的结果是整洁的。

##### 4.1.4.7.py

### 7、属性获取

还是用@符号就可以。

##### 4.1.4.8.py

注意于属性匹配的区别

### 8、属性多值匹配

有时候，某些节点的某个属性可能有多个值

例如：

<li class = 'li li-first'><a href="link.html">first item </a></li>

这里li节点的class属性有两个值li和li-first，这是需要用到contians（）函数

//li[contains(@class,"li-second")]/a/text()

##### 4.1.4.9.py

### 9、多属性匹配

多个属性确定一个节点，

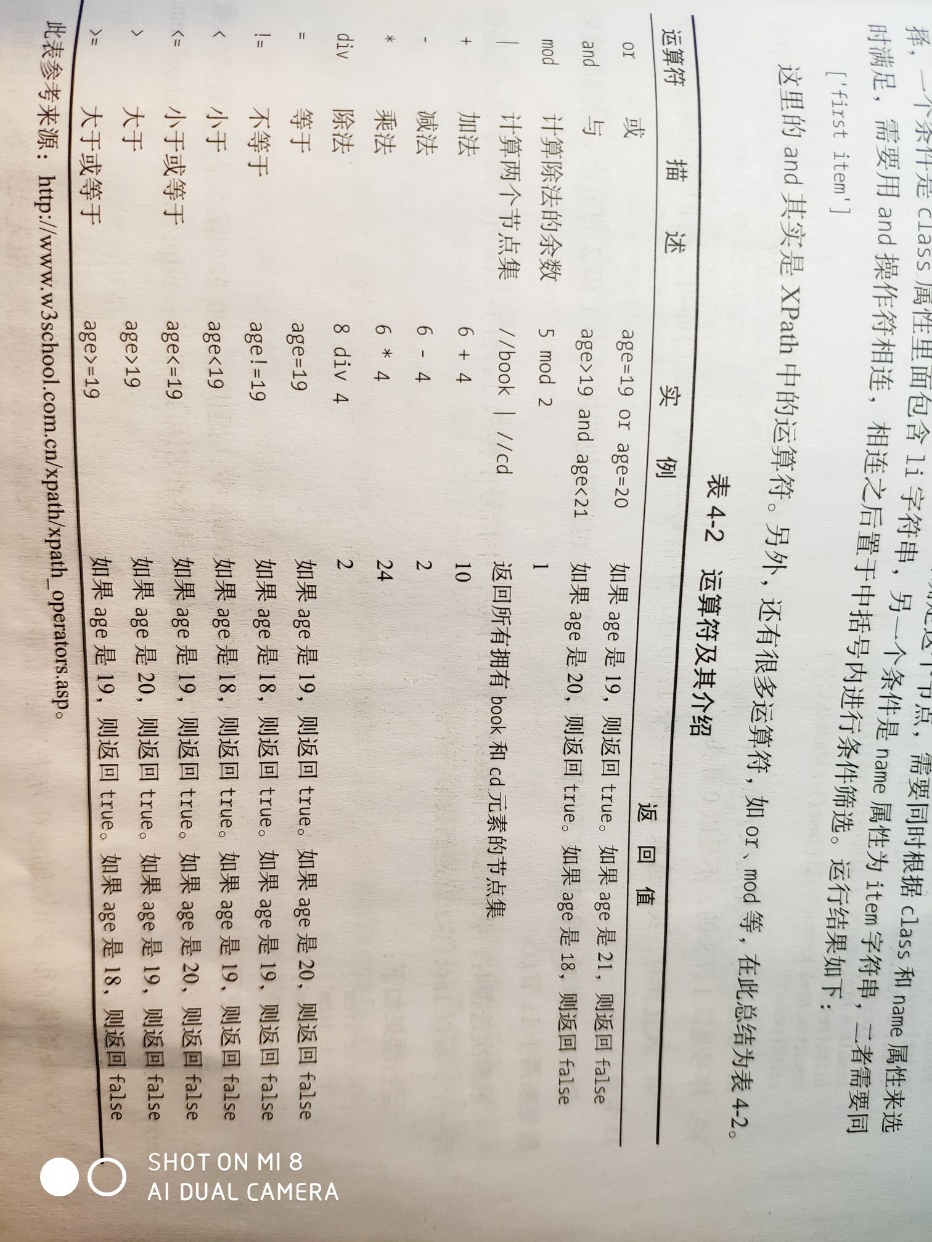
例如：

<li class = 'li li-first' name = 'item1'><a href="link.html">first item </a></li>  
<li class = 'li li-second' name = 'item2'><a href="link.html">second item </a></li>

要确定节点，需要同时根据class和name属性来选择，一个条件是class属性里面包含li字符串，另一个条件是name属性为item字符串，二者需要同时满足，需要使用and操作符，相连之后，置于中括号内进行筛选。

##### 4.1.4.10.py

这里的and其实是xpath的运算符，另外还有很多，例如，or、mod 等



### 10、按序选择

有时候，我们在选择的时候某些属性可能同时匹配了多个节点，到那时只想要其中的某个节点，但是只想要其中的某个节点，例如第二个节点或者最后一个节点。

利用中括号传入索引的方法来获取特定次序的节点。

##### 4.1.4.11.py

我们使用了last()，position()等函数，在xpath()中提供了100多个函数，包括存取，数值，字符串，逻辑，节点，序列等处理功能，具体参考，http://www.w3school.com.cn/xpath/xpath\_functions.asp

### 11、节点轴选择

Xpath 提供了很多节点轴选择方法，包括获取子元素，兄弟元素，父元素，祖先元素等

例如：

##### 4.1.4.12.py

更多轴的用法参考：http://www.w3school.com.cn/xpath/xpath\_axes.asp