1、什么是xpath

XPath (XML Path Language) 是一门在 HTML\XML 文档中查找信息的**语言**，可用来在 HTML\XML 文档中对**元素和属性进行遍历。**

W3School官方文档：http://www.w3school.com.cn/xpath/index.asp

2、认识xml

知识点：

- html和xml的区别
- xml中各个元素的的关系和属性

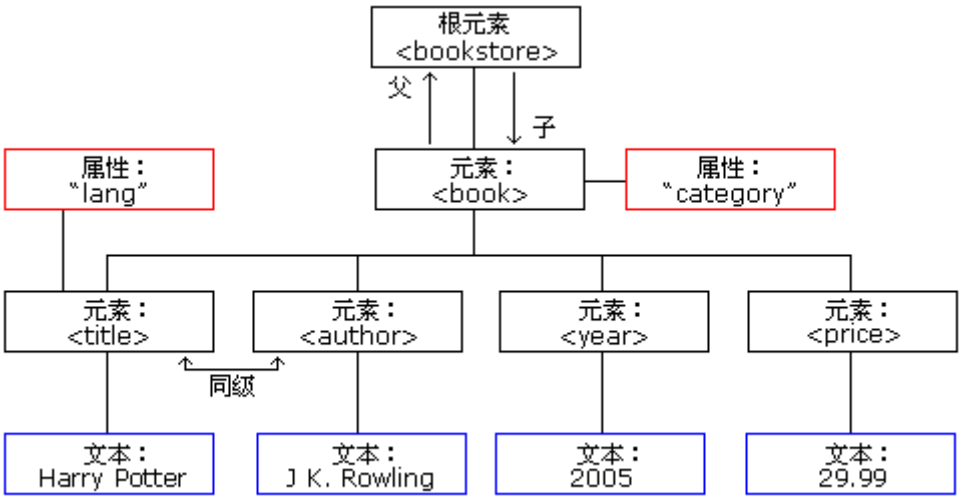
2、1 html和xml的区别

数据格式	描述	设计目标
XML	Extensible Markup Language （可扩展标记语言）	被设计为传输和存储数据，其焦点是数据的内容。
HTML	HyperText Markup Language （超文本标记语言）	显示数据以及如何更好显示数据。

2、2 xml的树结构

```
<bookstore>
<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

上面的xml内容可以表示为下面的树结构



上面的这种结构关系在xpath被进一步细化

3、xpath的节点关系

知识点：

- 认识xpath中的节点
- 了解xpath中节点之间的关系
- 每个html的标签我们都称之为节点。（根节点、子节点、同级节点）

4、xpath语法

XPath 使用路径表达式来选取 XML 文档中的节点或者节点集。这些路径表达式和我们在常规的**电脑文件系统中看到的表达式**非常相似。

下面列出了最有用的表达式：

表达式	描述
nodename	选中该元素。
/	从根节点选取、或者是元素和元素间的过渡。
//	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。跨节点获取标签
.	选取当前节点。
..	选取当前节点的父节点。
@	定位, 选取属性值
text()	选取文本。

在下面的表格中，我们已列出了一些路径表达式以及表达式的结果：

路径表达式	结果
bookstore	选择bookstore元素。
/bookstore	选取根元素 bookstore。注释：假如路径起始于正斜杠(/)，则此路径始终代表到某元素的绝对路径！
bookstore/book	选取属于 bookstore 的子元素的所有 book 元素。
//book	选取所有 book 子元素，而不管它们在文档中的位置。
bookstore//book	选择属于 bookstore 元素的后代的所有 book 元素，而不管它们位于 bookstore 之下的什么位置。
//book/title/@lang	选择所有的book下面的title中的lang属性的值。
//book/title/text()	选择所有的book下面的title的文本。

- 选取未知节点

通配符	描述
*	匹配任何元素节点。
@*	匹配任何属性节点。
node()	匹配任何类型的节点。

在下面的表格中，我们列出了一些路径表达式，以及这些表达式的结果：

路径表达式	结果
/bookstore/*	选取 bookstore 元素的所有子元素。
//*	选取文档中的所有元素。
//title[@*]	选取所有带有属性的 title 元素。

- 案例

```
        <div>
            <ul>
                <li class="item-1">
                    <a href="link1.html">第一个</a>
                </li>

                <li class="item-2">
                    <a href="link2.html">第二个</a>
                </li>

                <li class="item-3">
                    <a href="link3.html">第三个</a>
                </li>

                <li class="item-4">
                    <a href="link4.html">第四个</a>
                </li>

                <li class="item-5">
                    <a href="link5.html">第五个</a>
                </li>
            </ul>
        </div>
```

```
import parsel  # str --> Selector对象    具有xpath方法    提取到的数据返回一个列表

html_str = """
        <div>
            <ul>
                <li class="item-1">
                    <a href="link1.html">第一个</a>
                </li>

                <li class="item-2">
                    <a href="link2.html">第二个</a>
                </li>

                <li class="item-3">
                    <a href="link3.html">第三个</a>
                </li>

                <li class="item-4">
                    <a href="link4.html">第四个</a>
                </li>

                <li class="item-5">
                    <a href="link5.html">第五个</a>
                </li>
            </ul>
        </div>
    """
# 1、转换数据类型
```

```
# data = parsel.Selector(html_str).extract()  # parsel能够把缺失的html标签补充完成
data = parsel.Selector(html_str)  # parsel能够把缺失的html标签补充完成
# 2、解析数据--list类型
# print(data)
# 2、1 从根节点开始，获取所有<a>标签
result = data.xpath('/html/body/div/ul/li/a').extract()
# 2、2 跨节点获取所有<a>标签
result = data.xpath('//a').extract()
# 2、3 选取当前节点    使用场景：需要对选取的标签的下一级标签进行多次提取
result = data.xpath('//ul')
result2 = result.xpath('./li').extract()    # 提取当前节点下的<li>标签
result3 = result.xpath('./li/a').extract()  # 提取当前节点下的<a>标签
# 2、4 选取当前节点的父节点,获取父节点的class属性值
result = data.xpath('//a')
result4 = result.xpath('../@class').extract()
# 2、5 获取第三个<li>标签的节点（两种方法）
result = data.xpath('//li[3]').extract()
result = data.xpath('//li')[2].extract()
# 2、6 通过定位属性的方法获取第四个<a>标签
result = data.xpath('//a[@href="link4.html"]').extract()
# 2、7 用属性定位标签，获取第四个<a>标签包裹的文本内容
result = data.xpath('//a[@href="link4.html"]/text()').extract()
# 2、8 获取第五个<a>标签的href属性值
result = data.xpath('//li[5]/a/@href').extract()
# 了解 模糊查询
result = data.xpath('//li[contains(@class,"it")]').extract()
# 同时获取<li>标签的属性以及<a>标签的文本
# result = data.xpath('//li/@class|//a/text()').extract()

print(result)
```

- 如何选取多个标签?

  通过在路径表达式中使用“|”运算符，您可以选取若干个路径。（逻辑运算符）

6、小结

1. xpath的概述XPath (XML Path Language),解析查找提取信息的语言
2. xpath的节点关系:根节点,子节点,同级节点
3. xpath的重点语法获取任意节点:`//`
4. xpath的重点语法根据属性获取节点:`标签[@属性 = '值']`
5. xpath中获取节点的文本：`text()`
6. xpath的获取节点属性值:`@属性名`

@拓展知识

转义字符

在需要在字符中使用特殊字符时，python 用反斜杠转义字符。如下表：

转义字符	描述
\	(在行尾时)续行符
\	反斜杠符号
`\'`	单引号
`\"`	双引号
\a	响铃
\b	退格(Backspace)
\e	转义
\000	空
\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车
\f	换页
\oyy	八进制数，yy代表的字符，例如：\o12代表换行
\xyy	十六进制数，yy代表的字符，例如：\x0a代表换行
\other	其它的字符以普通格式输出

原始字符串

由于字符串中的反斜线都有特殊的作用，因此当字符串中包含反斜线时，就需要使用转义字符 \ 对字符串中包含的每个 "" 进行转义。

比如说，我们要写一个关于 Windows 路径 G:\publish\codes\02\2.4 这样的字符串，如果在 Python 程序中直接这样写肯定是不行的，需要使用 \ 转义字符，对字符串中每个 '' 进行转义，即写成 G:\publish\codes\02\2.4 这种形式才行。

有没有觉得这种写法很啰嗦，有没有更好的解决办法呢？答案是肯定的，借助于原始字符串可以很好地解决这个问题。

原始字符串以“r”开头，它不会把反斜线当成特殊字符。因此，上面的 Windows 路径可直接写成如下这种形式：

```
# 原始字符串包含的引号，同样需要转义
s2 = r'"Let\'s go", said Charlie'
print(s2)
```