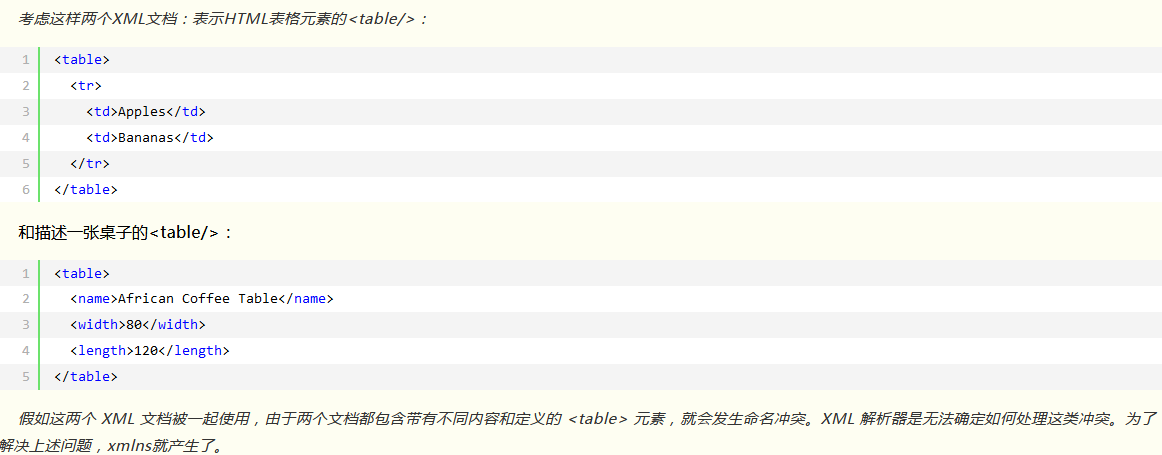
## 什么是xml的命名空间？

其实xml的命名空间可以类比java中的包。xml命名空间中定义的是标签名和属性名，是由国际化资源标识符 (IRI) 标识的 XML 元素和属性集合，该集合通常称作 XML“词汇”。它的作用就是为了防止和同名元素的冲突。

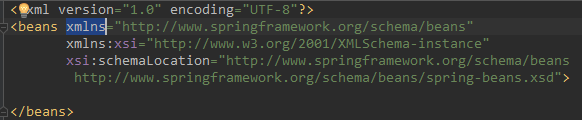


## xml命名空间的声明(xmlns)

命名空间被声明为元素的属性，虽然我们一般将命名空间的声明放在根元素上，但是其实命名空间的声明可以放在xml文档的任何一个元素标签中。

命名空间的范围应用于声明元素中的所有内容，直到被具有相同前缀名称的其他命名空间覆盖。

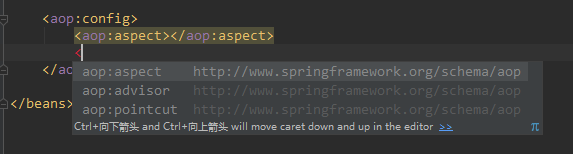
例如，这里是在beans中声明，那么对于该标签内都有效。



我们注意到xmlns其实可以译为xml命名空间，xml namespace的缩写。看下面的例子：



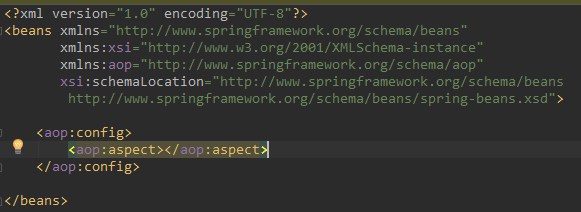
通过xmlns将前缀aop和命名空间http://…绑定在了一起，这里的前缀其实是可以自定义的，只不过我们都约定俗成的使用罢了。



我们通过使用aop前缀，可以让xml分析器知道后面的元素，如config，aspect都在那个URL中定义，然后寻找xsd文件校验，所以只要一打就可以弹出来，说明分析器已经知道去哪里找。

### 单个默认的命名空间

这里再次引用第一张图片：

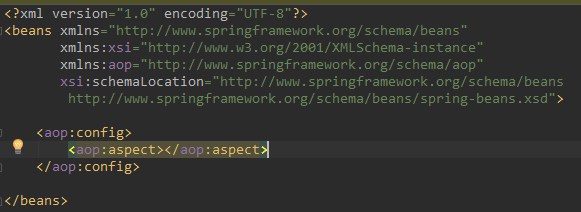


看见beans标签中，xmlns并没有给命名空间绑定任何前缀，而且可以看出beans是没有使用命名空间前缀的，这个就是默认的命名空间。若声明范围内没有显示前缀的标签都会被隐式限定。

正是因为比较常用，所以干脆就设置为默认，不用写前缀了。

需要注意到是默认的只能有一个。

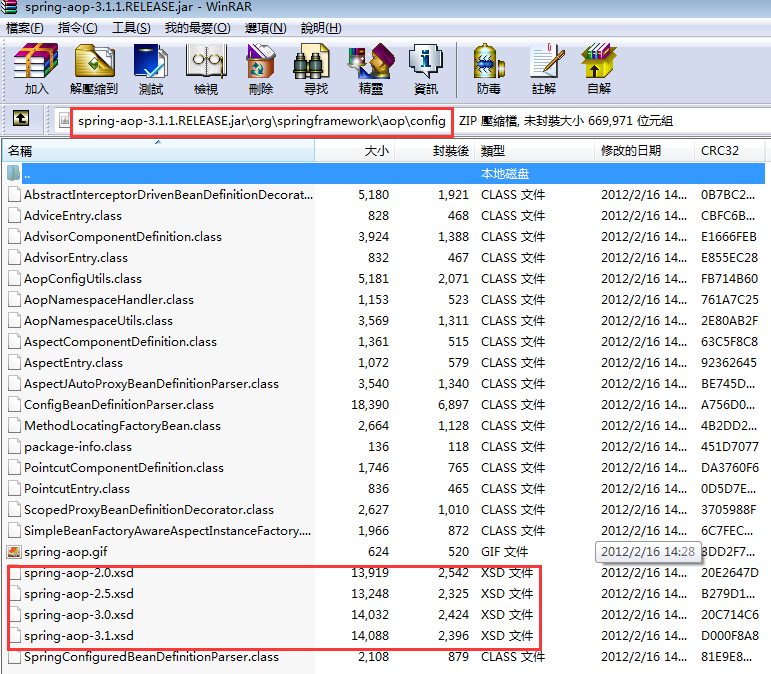
## xsi.schemaLocation有何作用？



看见这个属性带有xsi前缀，所以后面的schemaLocation的定义是在xmlns:xsi命名空间中的。

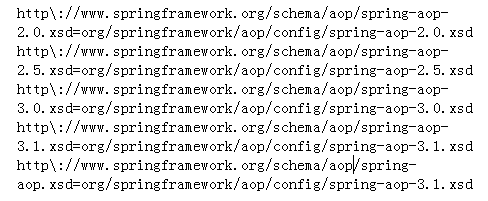
它定义了XML Namespace和对应的XSD(Xml Schema Definition)文档的位置关系。它的值由一个或多个URI引用对组成，两个URI之间以空白符分隔（空格和换行均可）。第一个URI是定义的 XML Namespace的值，第二个URI给出Schema文档的位置。

而Spring引入xsd文件是用来检验xml文件的格式用的。所以在Spring启动时是要加载XSD文件来验证xml文件的，所以如果有的时候断网了，或者一些开源软件切换域名，那么就很容易碰到应用启动不了。为了防止这种情况，Spring提供了一种机制，默认从本地加载XSD文件。



图片中的xsd就是本地的。

而在META-INF文件夹中的spring.schema中：



可以看见，和本地的路径是一一对应的。

### Spring中的xsd文件带版本号与不带版本号的区别

Spring的配置里,最好不要配置xsd文件的版本号。

因为如果没有配置版本号，取的就是当前jar里的XSD文件，减少了各种风险。其实，最大的好处是，当你升级了Spring的jar，该配置文件的XSD声明部分也不用改，会自动找到最新的本地jar里的XSD声明。Spring做了特殊处理，保证断网的情况下不写XSD版本号，能找到本地的jar文件。

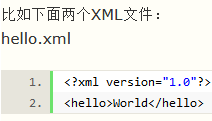
上面那张图的最后一个，是不带版本号的，默认去jar包中寻找最新的xsd文件。

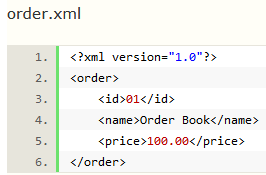
## 什么是XML Schema？

XML Schema 的作用是定义 XML 文档的合法构建模块(也就是文档的结构)，类似 DTD，是DTD的替代者。换句话说，Schema是一种标准，用来规定xml文档可以包含哪些标签，属性等。

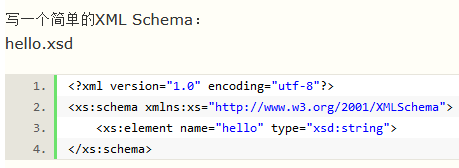
* 定义可出现在文档中的元素
* 定义可出现在文档中的属性
* 定义哪个元素是子元素
* 定义子元素的次序
* 定义子元素的数目
* 定义元素是否为空，或者是否可包含文本
* 定义元素和属性的数据类型
* 定义元素和属性的默认值以及固定值

举个例子：





怎么处理这些文件？通过XML Schema来处理。



首先schema是在xs命名空间中的，element元素中的name属性描述的是xml文档中的节点，type表示节点值的类型