# 元素

## <!ELEMENT>

此标签定义了一个标签

例如: <!ELEMENT root (#PCDATA)>

定义了一个下方的<root>标签, 类型是PCDATA

### PCDATA

PCDATA 的意思是被解析的字符数据（parsed character data）。可把字符数据想象为 XML 元素的开始标签与结束标签之间的文本。PCDATA 是会被解析器解析的文本。这些文本将被解析器检查实体以及标记。文本中的标签会被当作标记来处理，而实体会被展开。不过，被解析的字符数据不应当包含任何 &、< 或者 > 字符；需要使用 &amp;、&lt; 以及 &gt; 实体来分别替换它们。只有 PCDATA 的元素通过圆括号中的 #PCDATA 进行声明,例如<!ELEMENT meta (#PCDATA)> 定义的就是

<meta></meta>中间的数据会被解析 但中间不可加其他表情

<!ELEMENT body ANY> 例如 html的body

里面可以任意加数据或者其他表情

### CDATA

CDATA 的意思是字符数据（character data）CDATA 是不会被解析器解析的文本。在这些文本中的标签不会被当作标记来对待，其中的实体也不会被展开

这也是为何在xml文件中,需要使用转义符的原因, 或者使用CDATA的原因 即 使得内部的符号不解析,仅当做普通字符处理,这里一般在属性中使用,例如我们再属性中定义任何值都是可以的, 因为会被当做字符串

例如

<!ELEMENT property EMPTY>

<!ATTLIST property

name CDATA #REQUIRED

value CDATA #REQUIRED

>

定义的就是

<property name = “1>!$$” value = “2”/>

其中name和value都是必填的, name中的任何字符都会被当做字符串处理

### EMPTY

表示此元素为空 例如<br/>,<img/>

此时在dtd中定义为

<!ELEMENT img EMPTY>

<!ELEMENT br EMPTY>

### ANY

可以包含任何可解析的数据组合.

例如<!ELEMENT body ANY> 这里类型与html的body 内部可以放任何字符串以及标签, 不同的是 这里需要用到转义符

### 符号说明

1.\* <!ELEMENT root (head\*)>表示root下head可以出现任意次数

2.什么都不加<!ELEMENT root (head)> 表示root下head必须出现且仅出现一次

例如maven配置文件.依赖下面必须指定aid与gid,则需要这样定义.

<!ELEMENT depency (aid,gid)>

<!ELEMENT aid EMPTY>

<!ELEMENT gid EMPTY>

3.? <!ELEMENT root (head?)>表示root下head最多可以出现1次

4.+ <!ELEMENT root (head+)>表示root下head至少出现一次

5.|类似于或者, <!ELEMENT mapper(sql\*,(select | update | delete| insert))> 表示在mapper下,sql可以出现任意次数,必须出现select等等这几个的任意一个

# 属性

<!ATTLIST 名称 属性值 要求>

定义一个ELEMENT的属性

例如mybatis中的select

<!ELEMENT select (一些标签)>

<!ATTLIST select

id CDATA #REQUIRED

parameterType #IMPLIED

parameterMap #IMPLIED

resultType CDATA #IMPLIED

resultMap CDATA #IMPLIED

…

(

假如有readOnly的话,应该如此定义

readOnly CDATA ( ture | false ) “false”默认值是false

readOnly CDATA ( ture | false ) #IMPLIED无默认值

)

>

如果使用属性作为数据容器，最终的XML文档将难以阅读和维护。 尝试使用元素来描述数据。只有在提供的数据是不相关信息时才建议使用属性。

例如

<data time = “2018/12/12”/>与

<data>

<year>2018</year>

<month>12</month>

<day>12</day>

<data>

这两个表达的信息是相同的

而在存储不是数据的一部分的时候, 还是使用属性比较好, 例如select里面的属性 id, resultType 因为select这个标签内主要是为了写sql语句,id是为了做唯一标识来映射mapper方法.故而应该放在属性中

# 实体

内部实体

<!ENTYITY 实体名称 “属性值”>

例如

<!ENTITY author “oupaijizhengyi”>

使用时

<author>&author;<author>

显示的时候就会是 oupaijizhengyi

外部实体

<!ENTITY 实体名称 SYSTEM “url”>这里的url就链接的是一个系统内dtd文件, 如果上方定义的内部实体存储在myDtd.dtd中, 那么这里要引用的时候就可以定义

<!ENTITY writer SYSTEM “myDtd.dtd”>当然这里是在系统里, 如果在网络的话,存储在我个人的网站上

<!ENTITY writer SYSTEM “http://oupaijizhengyi.org/dtd/myDtd.dtd”>

使用时

<name>&writer;</name>使用的效果与上方相同

# 使用

<!DOCTYPE 基础元素名称 SYSTEM “URL”>

或者直接在这里定义

<!DOCTYPE html[

<!ELEMENT head ANY>

]>

放入需要约束的xml文件顶部即可