

## XML 元素 vs. 属性

在XML中，并没有规定何时使用属性，以及何时使用子元素。

### 使用元素 vs. 属性

数据可以存储在子元素或属性。

让我们来看下这些实例:

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

在第一个例子中"sex"是一个属性。在后面一个例子中，"sex"是一个子元素。但是两者都提供了相同的信息。

没有特别规定何时使用属性，以及何时使用子元素。我的经验是在 HTML 中多使用属性，但在XML中，使用子元素，会感觉更像数据信息。

## 我喜欢的方式

### 我喜欢在子元素中存储数据

下面的三个XML文档包含完全相同的信息：

本例中使用"date"属性：

```
<note date="12/11/2002">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

本例中使用"date"元素：

```
<note>
  <date>12/11/2002</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

本例中使用了扩展的"date" 元素: (这是我最喜欢的方式):

```
<note>
  <date>
    <day>12</day>
    <month>11</month>
    <year>2002</year>
```

```
</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

## 避免使用属性？

你应该避免使用属性？

一些属性具有以下问题：

- 属性不能包含多个值（子元素可以）
- 属性不容易扩展（为以后需求的变化）
- 属性无法描述结构（子元素可以）
- 属性更难以操纵程序代码
- 属性值是不容易测试，针对DTD

如果您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。只有在提供的信息是不相关信息时才建议使用属性。

不要这个样子结束（这不是XML应该使用的）：

```
<note day="12" month="11" year="2002"
to="Tove" from="Jani" heading="Reminder"
body="Don't forget me this weekend!">
</note>
```

## 一个属性规则的例外

规则总是有另外的

关于属性的规则我有一个例外情况。

有时我指定的 ID 应用了元素。这些 ID 应用可在HTML中的很多相同的情况下可作为 NAME 或者 ID 属性来访问 XML 元素。以下实例展示了这种方式：

```
<messages>
<note id="p501">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>

<note id="p502">
  <to>Jani</to>
  <from>Tove</from>
  <heading>Re: Reminder</heading>
  <body>I will not!</body>
</note>
</messages>
```

以上实例的XML文件中，ID是只是一个计数器，或一个唯一的标识符，来识别不同的音符，而不是作为数据的一部分。

在这里我想说的是，元数据（关于数据的数据）应当存储为属性，而数据本身应当存储为元素。

← DTD 属性	DTD 实体 →
----------	----------

 点我分享笔记

---