

## 6.5 将字典转换为XML

### 问题

你想使用一个Python字典存储数据，并将它转换成XML格式。

### 解决方案

尽管 `xml.etree.ElementTree` 库通常用来做解析工作，其实它也可以创建XML文档。例如，考虑如下这个函数：

```
from xml.etree.ElementTree import Element

def dict_to_xml(tag, d):
    """
    Turn a simple dict of key/value pairs into XML
    """
    elem = Element(tag)
    for key, val in d.items():
        child = Element(key)
        child.text = str(val)
        elem.append(child)
    return elem
```

下面是一个使用例子：

```
>>> s = { 'name': 'GOOG', 'shares': 100, 'price': 490.1 }
>>> e = dict_to_xml('stock', s)
>>> e
<Element 'stock' at 0x1004b64c8>
>>>
```

转换结果是一个 `Element` 实例。对于I/O操作，使用 `xml.etree.ElementTree` 中的 `tostring()` 函数很容易就能将它转换成一个字节字符串。例如：

```
>>> from xml.etree.ElementTree import tostring
>>> tostring(e)
b'<stock><price>490.1</price><shares>100</shares><name>GOOG</name></stock>'
>>>
```

如果你想给某个元素添加属性值，可以使用 `set()` 方法：

```
>>> e.set('_id', '1234')
>>> tostring(e)
b'<stock _id="1234"><price>490.1</price><shares>100</shares><name>GOOG</name></stock>'
>>>
```

如果你还想保持元素的顺序，可以考虑构造一个 `OrderedDict` 来代替一个普通的字典。请参考1.7小节。

### 讨论

当创建XML的时候，你被限制只能构造字符串类型的值。例如：

```
def dict_to_xml_str(tag, d):
    """
    Turn a simple dict of key/value pairs into XML
    """
    parts = ['<{}>'.format(tag)]
    for key, val in d.items():
        parts.append('<{}>{}</{}>'.format(key, val))
    parts.append('</{}>'.format(tag))
    return ''.join(parts)
```

问题是如果你手动去构造的时候可能会碰到一些麻烦。例如，当字典的值中包含一些特殊字符的时候会怎样呢？

```
>>> d = { 'name' : '<spam>' }

>>> # String creation
>>> dict_to_xml_str('item', d)
'<item><name><spam></name></item>'

>>> # Proper XML creation
>>> e = dict_to_xml('item', d)
>>> tostring(e)
b'<item><name>&lt;spam&gt;</name></item>'
>>>
```

注意到程序的后面那个例子中，字符‘<’和‘>’被替换成了 `&lt;` 和 `&gt;`。

下面仅供参考，如果你需要手动去转换这些字符，可以使用 `xml.sax.saxutils` 中的 `escape()` 和 `unescape()` 函数。例如：

```
>>> from xml.sax.saxutils import escape, unescape
>>> escape('<spam>')
'&lt;spam&gt;'
>>> unescape(_)
'<spam>'
>>>
```

除了能创建正确的输出外，还有另外一个原因推荐你创建 `Element` 实例而不是字符串，那就是使用字符串组合构造一个更大的文档并不是那么容易。而 `Element` 实例可以不用考虑解析XML文本的情况下通过多种方式被处理。也就是说，你可以在一个高级数据结构上完成你所有的操作，并在最后以字符串的形式将其输出。