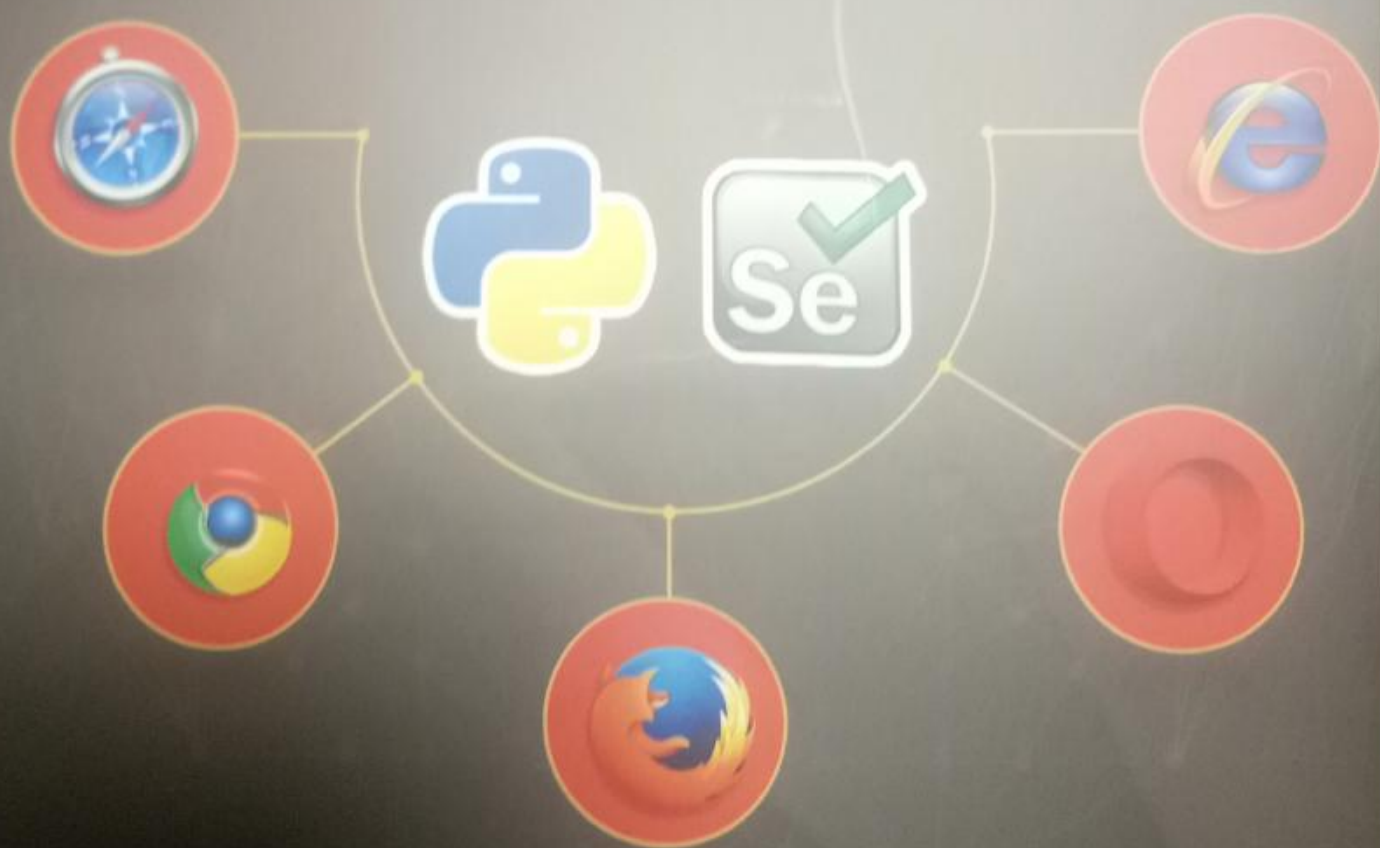


Selenium WebDriver

自动化测试完全指南

刘傲凡 著



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

7.3 将字典转化为 XML

XML (eXtensible Markup Language, 可扩展标记语言) 是一种灵活的标记语言。它的一系列编码转换的规则, 它的设计理念是让信息可以同时被人和机器阅读。文件占用的空间比单纯的二进制文件要多很多, 但是这种可扩展标记语言更易于阅读。它的这种特性使其易于在任何语言、程序中读写数据。和 HTML 相比, XML 格式的数据在存储空间的有效比重上不太占优势。

那么, 什么是 XML 呢? 我们来看一个简单的例子。

```
<?xml version="1.0" encoding="utf-8"?>
<dict>
  <name>zhang_san</name>
  <gender>female</gender>
  <phone_number>01234567890</phone_number>
  <note><note></note>
</dict>
```

它的格式和 HTML 很相像, 大标签中有小标签, 以此来对文本进行不同的标记。

在解析 XML 的时候, 我们主要会用到两个方法——Element 和 tostring。前者是一个 XML 文件, 后者则用于将 XML 文件转换为可以被读取、理解的字符串形式。

在以下代码中, 我们首先提供了一个字典, 然后写了一个可以普遍使用的函数, 将其字典转换为 XML。接下来输出 XML。第一次输出结果返回的是一个内容对象, 其内容存储在这里。第二次输出将其转换为了字符串, 接下来我们对最外层设置 ID 并再次输出。

```
from xml.etree.ElementTree import tostring
from xml.etree.ElementTree import Element

dict = {"name": "zhang_san", "gender": "female", "phone_number": "01234567890", "note": "note"}

# 写一个转换函数
def convert(tag_name, dict):
    # 建立一个 XML
    elem = Element(tag_name)
    # 通过双值循环取出字典中的键和值
```

```
for key, value in dict.items():
    # 输出
    print(key, ":", value)

    # 相当于为键创建一个标签
    xml_elem = Element(key)
    # 将键的值赋给这个标签
    xml_elem.text = str(value)

    # 添加到 XML 文件中
    elem.append(xml_elem)

return elem
```

```
# 字典转换
dict_convert = convert("test", dict)
print("-----")
print(dict_convert)

# 转换为可读的标准 XML
print(tostring(dict_convert))
# 设置最外层标签的属性
dict_convert.set('id_number', '01234567890')
# 再次输出
print(tostring(dict_convert))
```

输出结果如下。

```
name : zhang_san
gender : female
phone_number : 01234567890
note : note

<Element 'test' at 0x02C5DED0>
b'<test><name>zhang_san</name><gender>female</gender><phone_number>01234567890</phone_number><note>note</note></test>'
b'<test id_number="01234567890"><name>zhang_san</name><gender>female</gender><phone_number>01234567890</phone_number><note>note</note></test>'
```

0x02C5DED0 这是一个十六进制的内存地址, 而第二次输出和第三次输出的区别就在于 id_number="01234567890", 这是设置 ID 后修改的内容。