DOM xml para Python

Es un lenguaje API del Consorcio *World Wide Web* (W3C) para acceder y modificar documentos XML. Una implementación del DOM presenta los documento XML como un árbol, o permite al código cliente construir dichas estructuras desde cero para luego darles acceso a la estructura a través de un conjunto de objetos que implementaron interfaces conocidas.

Algunas aplicaciones son imposibles en un modelo orientado a eventos sin acceso a un árbol. Por supuesto que puedes construir algún tipo de árbol por tu cuenta en eventos SAX, pero el DOM te evita escribir ese código. El DOM es una representación de árbol estándar para datos XML.

Una vez que tengas un objeto del documento del DOM, puedes acceder a las partes de tu documento XML a través de sus propiedades y métodos. Estas propiedades están definidas en la especificación del DOM; está porción del manual describe la interpretación de la especificación en Python.

Ejemplos:

Uso del módulo xml-dom:

1) xml.dom.registerDOMImplementation(name, factory)

Registra la función factory con el nombre name. La función fábrica (factory) debe retornar un objeto que implemente la interfaz DOMImplementation. La función fábrica puede retornar el mismo objeto cada vez que se llame, o uno nuevo por cada llamada, según sea apropiado para la implementación específica.

2) xml.dom.getDOMImplementation(name=None, features=())

Retorna una implementación del DOM apropiada. El name es o bien conocido, el nombre del módulo de una implementación DOM, o None. Si no es None importa el módulo correspondiente y retorna un objecto DomImplementation si la importación tiene éxito.

3) xml.dom.EMPTY_NAMESPACE

El valor usado para indicar que ningún espacio de nombres es asociado con un nodo en el DOM. Se encuentra típicamente con el namespaceURI de un nodo, o usado como el parámetro namespaceURI para un método específico del namespace.

XPath XML para Python.

Xpath es un módulo que es parte de la librería xml.etree.ElementTree por lo general la misma se importa de la siguiente manera:

import xml.etree.ElementTree as ET

Xpath provee una serie de expresiones para localizar elementos en un árbol, su finalidad es proporcionar un conjunto de sintaxis, por lo que debido a su limitado alcance no se considera un motor en si mismo.

Ejemplo: aplicación de xPath.

3)

```
import xml.etree.ElementTree as ET

root = ET.fromstring(docxml)

# Elementos de nivel superior
root.findall(".")

# todos los hijos de neighbor o nietos de country en el nivel superior
root.findall("./country/neighbor")

# Nodos xml con name='Singapore' que sean hijos de 'year'
root.findall(".//year/..[@name='Singapore']")

# nodos 'year' que son hijos de etiquetas xml cuyo name='Singapore'
root.findall(".//*[@name='Singapore']/year")

# todos los nodos 'neighbor'que son el segundo hijo de su padre
root.findall(".//neighbor[2]")
```

nos permite extraer facilmente partes del xml haciendo referencia a su ubicación nodal representada a forma de path, lo que nos hace una sintaxis familiarmente sencilla a la hora de construir un paser xml.

Ejemplo de uso de Xpath dentro de una sentencia xml:

4)

```
<xpath expr="//field[@name]='is_done'" position="before">
    <field name="date_deadline" />
</xpath>
```