Laboratorio Introducción a la Programación y Computación 2

– Investigación DOM XML, xPath XML para Python --

Joaquin Emmanuel Aldair Coromac Huezo

201903873

xml.dom — El API del Modelo de Objetos del Documento

El Modelo de Objetos del Documento, o «DOM» por sus siglas en inglés, 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.

El DOM es extremadamente útil para aplicaciones de acceso directo. SAX sólo te permite la vista de una parte del documento a la vez. Si estás mirando un elemento SAX, no tienes acceso a otro. Si estás viendo un nodo de texto, no tienes acceso al elemento contendor. Cuando desarrollas una aplicación SAX, necesitas registrar la posición de tu programa en el documento en algún lado de tu código. SAX no lo hace por ti. Además, desafortunadamente no podrás mirar hacia adelante (look ahead) en el documento XML.

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.

El Modelo de Objetos del Documento es definido por el W3C en fases, o «niveles» en su terminología. El mapeado de Python de la API está basado en la recomendación del DOM nivel 2.

Las aplicaciones DOM típicamente empiezan al diseccionar (parse) el XML en un DOM. Cómo esto funciona no está incluido en el DOM nivel 1, y el nivel 2 provee mejoras limitadas. Existe una clase objeto llamada DOMImplementation que da acceso a métodos de creación de Document, pero de ninguna forma da acceso a los constructores (builders) de reader/parser/Document de una forma independiente a la implementación. No hay una forma clara para acceder a estos método sin un objeto Document existente. En Python, cada implementación del DOM proporcionará una función getDOMImplementation(). El DOM de nivel 3 añade una especificación para Cargar(Load)/Guardar(Store), que define una interfaz al lector (reader), pero no está disponible aún en la librería estándar de Python.

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.

xPath XML

XPath es un estándar para el lenguaje de path en XML. Existe desde hace más de 15 años y se conocen 3 versiones.

con Xpath es posible identificar partes del documento XML y también ejecutar computos sobre esa data.

Expresiones (expresions)

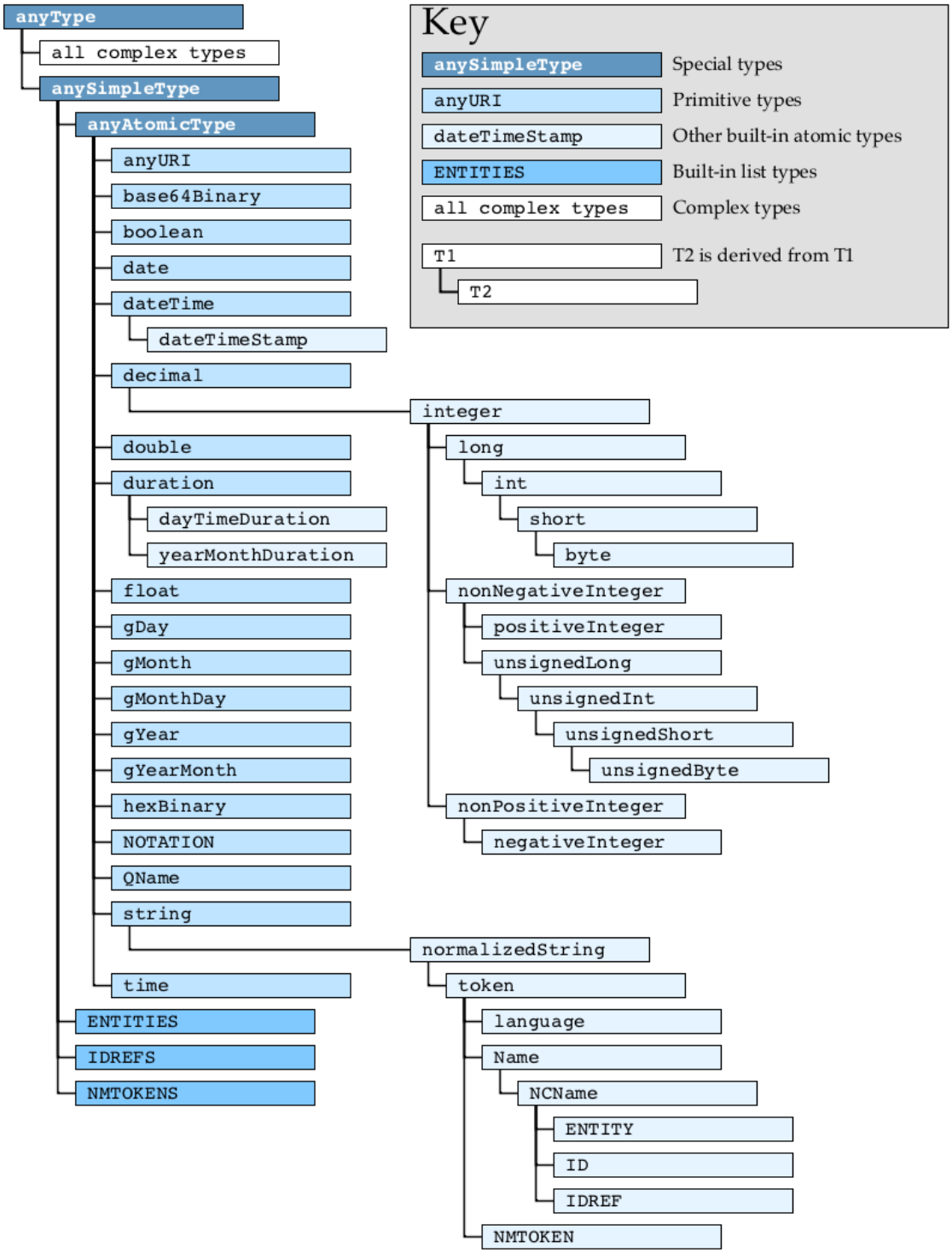
es el principal bloque de construcción en Xpath, una expresión es un simple string de caracteres unicode hecho de palabras claves, simbolos y operadores.

Secuencias (sequences)

Son muy importantes en Xpath ya que cada expresión retorna una secuencia. Así una secuencia es una simple colección de items que pueden ir desde cero o más de ellos. Una secuencia sin items o vacía se conoce como: secuencia vacía (empty sequence). Una secuencia con un item simple es conocida como secuencia simple (singleton sequence). a partir de Xpath 3.0 un item en una secuencia puede ser un nodo, un valor atómico (atomic value), o una función. De esta manera Xpath 3.0 define 7 tipos de nodos:

1. documento (document).
2. element.
3. attribute.
4. namespace.
5. procesing instruction.
6. comment.
7. text.

Un valor atómico es un valor permitido de un conjunto de valores para un tipo atómico en particular. Los tipos atómicos son todos los tipos que se derivan de forma directa o indirecta de **xs:anyAtomicType**.Como se especifica en el diagrama de herencia de tipos de datos para XML Schema 1.1:



La **función** es un nuevo tipo de item. En Xpath 2.0 un item en una secuencia solo podía ser un nodo o un valor atómico. En Xpath 3.0 puede ser un nodo, un valor atómico o una función.

Ubicación de la ruta (path)

Ubicar rutas es Xpath es muy similar a usar la notación de path para archivos en el sistema de archivos que usemos y al igual que este último tenemos rutas relativas y absolutas, donde una ruta absoluta siempre es evaluada desde la raiz del árbol “/” y una ruta relativa se referencia a un nodo en específico. la raiz del

documento “/” es el nivel más alto del XML y es el contenedor de todos los nodos que lo componen inclusive el mismo. Una ruta absoluta también puede empezar con: “//” (lo que se refiere a todos los subnodos de la raiz) lo que es distinto a “/” (se refiere a la raiz, todo el documento, incluyendo nodos, expresiones y/o funciones). Una ruta relativa es siempre evaluada desde el contexto de un nodo. Ejemplo:

office/employee/first_name[.=’John’]

El nodo contextual en el que esta evaluada está expresión es ‘company’ que es la raiz del archivo company_1.xml, el elemento padre es ‘office’. El elemento contextual puede cambiar durante la evaluación de la expresión como lo indica el carácter punto ‘.’ en este caso es usado para indicar el nodo contextual ‘first_name’ de esta forma se compara el valor de este nodo con el string ‘John’.

Steps

Una ruta de path contiene uno o más steps. Un step está compuesto de:

1. una axis. Es la primera parte de una ubicación step la que determina la dirección a navegar con respecto de un nodo particular. Existen 13 diferentes axisas pertenecientes a dos grupos: **forward axis** (estás retornan en el mismo orden que se encuentran en el documento XML) o **reverse axis (retornan de forma inversa al documento XML). Se usa “::” para separar la axisa especifica del node test. ***
2. un node test. Aparece inmediatamente después de la axis y pueden ser de tres tipos: por nombre, por kind o por tipo.
3. Cero o más predicates.

axis::node_test[predicate]

child::office[@location='Viena']

Está ultima expresión selecciona desde el nodo contextual todos los elementos hijos de ‘office’ que tengan un atributo llamado ‘location’ que a su vez sea igual a ‘Vienna’.

Algunas Forward Axis:

- child
- descendant
- attribute
- self
- desendant-or-self
- following-sibling
- following

- namespace

Algunas Reverse Axis:

- parent
- ancestor
- preceding-sibling
- preceding
- ancestor-or-self