



Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Transformación XML a SVG

Dennis Gabriela Coy Calderón

Universidad de Oviedo
Escuela de Ingeniería Informática
Software y Estándares para la Web

Descripción

El objetivo de este documento es explicar la transformación del archivo XML a un archivo SVG por medio del lenguaje de programación Python.

Estructura del archivo SVG

El archivo SVG desarrollado sigue la siguiente estructura:

```
<?xml version="1.0" encoding="UTF-8"?>

<svg width="2000" height="5270" style="overflow:visible " version="1.1"
xmlns="http://www.w3.org/2000/svg">

<rect x="..." y="..." width="..." height="..." style="fill:white;stroke:black;stroke-width:1" />

<text x="..." y="..." font-size="10" style="fill:black">nombre</text>

<text x="..." y="..." font-size="10" style="fill:purple">Descripción</text>

<text x="..." y="..." font-size="8" style="fill:purple">Datos Nacimiento</text>

<text x="..." y="..." font-size="8" style="fill:purple">Fecha: ...</text>

<text x="..." y="..." font-size="8" style="fill:purple">Lugar: ...</text>

<text x="..." y="..." font-size="8" style="fill:purple">Coordenadas: lat, lon, alt</text>

<text x="..." y="..." font-size="8" style="fill:purple">Datos Fallecimiento</text>

<text x="..." y="..." font-size="8" style="fill:purple">Fecha: ...</text>

<text x="..." y="..." font-size="8" style="fill:purple">Lugar: ...</text>

<text x="..." y="..." font-size="8" style="fill:purple">Coordenadas: lat, lon, alt</text>

<path d="M... ... L... ..." stroke="purple" stroke-width="2" fill="none" />

</svg>
```

La estructura anterior define una caja con el elemento rect dadas sus coordenadas y su altura y ancho y el texto que llevará dentro con los elementos text dadas sus coordenadas en x e y.

Construcción del código en Python

Para esta transformación se hizo uso de Python 3.8 mediante el entorno de desarrollo Spyder.

El código xml2svg.py utiliza la librería xml.etree.ElementTree para la conversión del archivo XML en su árbol y así poder recorrer sus elementos como nodos. Está compuesto de las siguientes funciones:

- datosXML(archivoXML)

Esta función recibe como parámetro el archivo XML para extraer sus datos y retornarlo como una lista de diccionarios, en la que cada diccionario será una persona y su información. Detalle:

Paso 1. Se hace uso de la función `parse(archivoXML)` para convertir el archivo XML en un árbol.

Paso 2. Se obtiene la raíz del árbol mediante la función `getroot()`

Paso 3. Con la función `iter(elemento)` se busca a partir de la raíz todos los elementos que sean del mismo tipo, en este caso, elemento será igual a una cadena de caracteres con el nombre que recibe personas en el árbol XML, es decir, `'{http://www.uniovi.es}persona'`. Todo esto dentro de un ciclo para que cuando encuentre todos esos elementos se pueda extraer la información de cada uno de ellos.

Paso 4. Extraer las coordenadas de cada persona. Para esto se indica `persona[0]` para que tome el primer hijo de persona que es datos y con el uso de la función `find(elemento)` encontrar las coordenadas de nacimiento, fallecimiento y el comentario. Además, se hace uso del atributo `.text` para obtener el valor del elemento.

Paso 5. Se guardan estos datos en una lista y posteriormente se cambian las posiciones de los diccionarios en esta lista para facilitar el posicionamiento en el documento, dependiendo de una variable global llamada `posicion` que contiene los datos de las posiciones en las que se pondrá cada diccionario.

- `prologoSVG(archivo)`

Esta función recibe como parámetro el archivo de salida para escribir el prólogo del documento.

- `escribirElemento(archivo, nombre, descripcion, x, y)`

Esta función recibe como parámetros el archivo de salida, la persona a escribir en el archivo de salida y las coordenadas `x` y `y` de partida para graficar la caja y escribir el texto correspondiente a los datos de la persona. Cabe resaltar que en descripción solo escribirá 33 caracteres para que no se salga de la caja.

- `añadirContenido(archivo, datos)`

Esta función recibe como parámetros el archivo de salida y la lista de diccionarios con la información de las personas para enviar los datos de cada persona a la función `escribirElemento`. Además, hace cambios en `x` y `y` para escribir cada línea de hijos y cada elemento en la posición adecuada.

- `añadirRelaciones(archivo)`

Esta función recibe como parámetro el archivo de salida para escribir las relaciones de cada nodo padre con su hijo.

Finalmente, en la función main() se recibe el archivo XML de entrada y se declara el nombre que recibirá el archivo de salida svg. Se hace un llamado del prologoSVG, añadirContenido, añadirRelaciones, se cierra la etiqueta svg y se muestra un mensaje de Creación exitosa.

SVG del árbol genealógico:

