

Introducción a la Programación con **PYTHON**



José Rodriguez

Unidad 7- Ficheros XML

UNIDAD 9 – GESTIÓN DE FICHEROS XML

Python dispone de un objeto específico para procesar ficheros XML. Este objeto es

`xml.etree.ElementTree`

Se trata de un puntero a un fichero XML físico.

El primer paso para procesar un fichero XML es, por tanto, crear una instancia de dicho objeto.

```
import xml.etree.ElementTree as ET
```

y asignarlo a un fichero físico traduciéndolo a python

```
tree = ET.parse('c:/dietetica.xml')
```

Admitamos que el fichero dietética.xml tiene la siguiente estructura

```
<dietetica>
    <Row>
        <IdProducto>1</IdProducto>
        <Nombre>Tienda Dharamsala</Nombre>
        <Proveedor>Exotic Liquids</Proveedor>
        <Categor_xfffd_a>Bebidas</Categor_xfffd_a>
        <Cantidad>10 cajas x 20 bolsas</Cantidad>
        <Precio>18,00</Precio>
        <Existencias>39</Existencias>
        <Pedido>0</Pedido>
        <Suspendido>FALSO</Suspendido>
    </Row>
    ...
</dietética>
```

El elemento raíz es la etiqueta `<dietética>` y está formado por una secuencias de elementos `<Row>` todos con la misma estructura que es la mostrada en el gráfico anterior.

El siguiente código mostrará el elemento raíz a partir del método `getroot()` que situa el puntero en el root del documento y después mostramos la propiedad `tag`.

```
import xml.etree.ElementTree as ET
tree = ET.parse('c:/dietetica.xml')
```

```
root = tree.getroot()
print(root.tag)
```

El resultado sería, evidentemente, dietética.

Utilizando el mismo código, podríamos mostrar otras propiedades del root del documento, por ejemplo, el número de elementos <Row> que forman el documento

```
import xml.etree.ElementTree as ET
tree = ET.parse('c:/dietetica.xml')
root = tree.getroot()
print("Hay " + str(len(root)) + " Registros")
```

No olvidemos que root apunta al raíz del documento (cuya etiqueta es dietetica) y que se trata de un array cuyos elementos son todos etiquetas de tipo <Row>. Este hecho lo podemos constatar si hacemos

```
producto=root[0].tag
```

producto es el primer elemento de la estructura <dietetica> y es por tanto una etiqueta <Row>

Una vez situado el puntero en un elemento determinado de la estructura XML podemos usar el método find() para buscar etiquetas dentro del objeto al que apunta puntero. Por ejemplo

```
import xml.etree.ElementTree as ET
tree = ET.parse('c:/dietetica.xml')
productos=tree.findall("Row")
print (productos[0].find("Nombre").text)
```

Analicemos el código. En primer lugar, tree apunta al documento. Productos forma un array con todos los objetos <Row> que cuelgan de tree. El primer elemento del array es productos[0]. Dentro de ese objeto buscamos la etiqueta <Nombre> y escribimos su propiedad text.

El resultado sería

Cerveza tibetana Barley

Usando esta estrategia podemos obtener un listado de todos los productos escribiendo su nombre. El código sería el siguiente

```
import xml.etree.ElementTree as ET
tree = ET.parse('c:/dietetica.xml')
root = tree.getroot()
productos=tree.findall("Row")
l=len(productos)
i=0
while i<l :
    print (productos[i].find("Nombre").text)
    i=i+1
```

y el resultado sería

Cerveza tibetana Barley
Sirope de regaliz
Especias Cajun del chef Anton
Mezcla Gumbo del chef Anton
Mermelada de grosellas de la abuela
.....

Podemos mezclar el bucle del ejercicio anterior con una estructura condicional para procesar los datos, por ejemplo, obtener un listado de todos los productos cuyas existencias sean superiores a 10 unidades.

```
i=0
while i<l :
    e=int(productos[i].find("Existencias").text)
    if e>10:
        print (productos[i].find("Nombre").text)
    i=i+1
```

En este ejemplo hemos convertido el campo text en un entero para poder hacer la comparación. Si el campo fuese un número con parte decimal, lo debemos convertir a float pero no olvides que el formato debe tener un punto decimal, no una coma

```
i=0
while i<l :
    e=float(productos[i].find("Precio").text)
    if e>10:
        print (productos[i].find("Nombre").text)
    i=i+1
```