

## Traductor XML a JSON

Implementar mediante JFlex y CUP un traductor de ficheros en XML a formato JSON. Estos son dos de los lenguajes de programación más utilizados para el intercambio de datos entre distintas aplicaciones en internet.

XML ([https://es.wikipedia.org/wiki/Extensible Markup Language](https://es.wikipedia.org/wiki/Extensible_Markup_Language)) es un lenguaje etiquetado de representación de información ampliamente utilizado en internet. Su principal característica es que representa la información mediante etiquetas o tags de la forma `<tag>valor</tag>`. Este formato, aunque sencillo, permite estructurar de forma muy eficiente multitud de tipos de datos como números, texto, listas, diccionarios, etc. En este traductor utilizaremos una versión reducida de este lenguaje.

Por otro lado, JSON es un formato sencillo (<https://es.wikipedia.org/wiki/JSON>) en el que los datos se representan dentro de llaves `{}`. Estos datos se organizan mediante un listado de pares `<clave>:<valor>`. Los distintos pares se separan mediante comas. JSON admite multitud de tipos de datos. Para simplificar, se seguirán las siguientes reglas:

- Las claves sólo podrán ser cadenas alfanuméricas. Estas podrán ir entre comillas dobles o sin ellas.
- Como valores se admitirán: palabras (entrecomillados o no), números enteros positivos (entrecomillados o no), otros JSON o listas.
- En el caso de las listas, estas se representarán entre corchetes y dentro podrán tener distintos valores (los cuales pueden ser de los mismos tipos admitidos que en JSON) separados por comas. Además se contempla la posibilidad de que exista una lista vacía `[]`.

Al igual que en el caso de los ficheros XML, se va a trabajar con una versión simplificada del lenguaje.

En general, vamos a representar la información utilizando como clave el tag del dato (sin los `<>`) y como valor, el mismo valor usado en el XML (con algunas excepciones).

Por ejemplo, la traducción del siguiente fichero sería la siguiente:

XML	JSON
<code>&lt;k&gt; valor &lt;/k&gt;</code>	<code>{k: valor}</code>

En los casos de ejemplo se presentarán las distintas normas a la hora de hacer la traducción<sup>1</sup>.

Se proporciona un analizador léxico implementado mediante JFlex, en el fichero `XML2JSON.flex`, la clase `XML2JSON.java` que contiene la implementación del método `main`. La lista de funciones disponibles se limita a los definidos en el analizador léxico.

Para compilar y ejecutar este traductor se usarán la secuencia:

```
cup XML2JSON.cup
jflex XML2JSON.flex
javac *.java
java XML2JSON <entrada>
```

Para la corrección de este ejercicio se entrega SOLAMENTE el fichero `XML2JSON.cup` por lo que cualquier modificación en cualquier otro fichero no se tendrá en cuenta al corregir.

El traductor debe de pintar por pantalla la traducción.

Se proporcionan diversos casos de prueba, con diversos aspectos que deben controlarse en la implementación.

---

<sup>1</sup> No se tendrán en cuenta los espacios en blanco, saltos de línea y demás separadores para las traducciones.



A continuación se presentan las siguientes consideraciones en la traducción. Para facilitar la lectura se muestran las salidas con saltos de línea y tabulaciones, las cuales no son necesarias de generar en la salida del programa.

Cada tag `<clave>valor</clave>` en XML, generar el par `<clave>: <valor>` en el JSON.

<pre>&lt;clave1&gt;valor1&lt;/clave1&gt; &lt;clave2&gt;valor2&lt;/clave2&gt; &lt;23&gt;valor3&lt;/23&gt;</pre>	<pre>{   clave1: valor1,   clave2: valor2,   23: valor3 }</pre>
--	---

Es posible anidar diferentes tags en XML, lo que genera distintos JSON anidados

<pre>&lt;k&gt;   &lt;subkey&gt;subvalue&lt;/subkey&gt;   &lt;subkey2&gt;subvalue2&lt;/subkey2&gt; &lt;/k&gt;</pre>	<pre>{   k: {     subkey: subvalue,     subkey2: subvalue2   } }</pre>
--	--

A pesar de no ser parte del estándar, es común emplear XML para generar listas de valores. Para ello, se utiliza la misma tag para los distintos valores, y todos ellos se agrupan dentro de una tag general escrita en plural (por simplificar, el plural será añadir una 's', es decir, no es necesario seguir las normas lingüísticas para los plurales). En JSON las listas se simplifican, ya que como clave se utilizará el tag de los elementos de la lista, y los valores se colocarán entre corchetes y separados por comas.

<pre>&lt;k&gt;v&lt;/k&gt; &lt;listas&gt;   &lt;lista&gt;1&lt;/lista&gt;   &lt;lista&gt;2&lt;/lista&gt;   &lt;lista&gt;3&lt;/lista&gt;   &lt;lista&gt;4&lt;/lista&gt; &lt;/listas&gt;</pre>	<pre>{   k: v,   listas: [1,2,3,4] }</pre>
--	--

Además, las listas pueden contener otros elementos además de números, como palabras, u otros tags (estos generarían nuevos JSONs).

<pre>&lt;ks&gt;   &lt;k&gt;1&lt;/k&gt;   &lt;k&gt;hola&lt;/k&gt;   &lt;k&gt;     &lt;a&gt;b&lt;/a&gt;     &lt;c&gt;12&lt;/c&gt;   &lt;/k&gt;   &lt;k&gt;     &lt;k1&gt;v1&lt;/k1&gt;   &lt;/k&gt; &lt;/ks&gt;</pre>	<pre>{   k: [     1,     hola,     { a: b,       c: 12 },     { k1: v1 }   ] }</pre>
---	--

#### NOTAS:

- La indentación de los ficheros XML y JSON no es relevante de cara a la sintaxis, y no se tiene en cuenta al corregir el ejercicio
- Ver otros ejemplos de ficheros XML y sus traducciones a JSON en el fichero auxiliar.