Desarrollo de Aplicaciones Web

XML y JSON

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

XML

Su nombre deriva de *eXtensible Markup Languaje*

- Como su nombre indica es un lenguaje de marcas, mantenido por W3C.

- Es un lenguaje muy útil para describir información estructurada.

- Facilitan el intercambio de información entre usuarios no humanos (máquinas)

- Existen herramientas desarrolladas por los principales fabricantes (Sun, Microsoft, Apache) que facilitan su uso.

XML

Archivos de propiedades *vs* archivos XML

Archivo de propiedades

Nombre= Valor
fontname= Times Roman
fontsize= 12
windowsize= 400 200
color= 0 50 100

Tienen jerarquía plana

Nombre= Valor
titulo.fontname= Times Roman
titulo.fontsize= 12
cuerpo.windowsize= 400 200
cuerpo.color= 0 50 100

No admiten repetición de valores

Nombre= Valor
menu.opción.1= Times Roman
menu.opción.2= Helvetica
menu.opción.3= Goudy Old Style

XML

Un archivo XML, permite expresar estructuras jerárquicas

```
<configuration>
        <title>
                <font>
                        <name>Helvetica</name>
                        <size>36</size>
                </font>
        </title>
        <body>
                <font>
                        <name>Times Roman</name>
                        <size>12</size>
                </font>
        </body>
        <window>
                <width>400</width>
                <height>200</height>
        </window>
        <color>
                <red>0</red>
                <green>50</green>
                <blue>100</blue>
        </color>
        <menu>
                <option>Times Roman</option>
                <option>Helvetica</option>
                <option>Goudy Old Style</option>
        </menu>
</configuration>
```

XML *vs* HTML

- XML distingue entre mayúsculas y minúsculas.
- En XML, siempre ha de incluirse el marcador final.
- En XML, los elementos con un único marcador han de acabar con “/>”.
- En XML, los valores de los atributos pueden ir encerrados entre comillas dobles o simples, pero ha de abrirse y cerrarse el mismo tipo de comillas.
- En XML, todos los atributos tienen que tener un valor.

XML. Estructura de un documento.

- Encabezado (opcional, recomendado)

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

- Definición de tipo de documento (opcional)

```
<!DOCTYPE web-app PUBLIC
        “-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN”
        http://java.sun.com/j2ee/dtds/web-app_2_2.dtd>
```

- Cuerpo (incluye elementos hijos, texto o ambas cosas. Son los elementos básicos)

```
<configuration>
        <title>
                <font>
                        <name>Helvetica</name>
                        <size>36</size>
                </font>
        </title>
</configuration>
```

XML. Consejos.

• Se debe evitar el contenido mixto (texto y elementos, al mismo nivel).

```
<font>
        Helvetica
        <size>36</size>
</font>
```

```
<font>
        <name>Helvetica</name>
        <size>36</size>
</font>
```

• Utilizar atributos únicamente para modificar la interpretación de un valor, no para especificar valores.

```
<font name="Helvetica" size="36 pt"/t>
```

```
<font>
        <name>Helvetica</name>
        <size unidad="pt">36</size>
</font>
```

JSON

Su nombre deriva de *JavaScript Object Notation*.

Formato para almacenamiento/intercambio de datos, creado inicialmente (1999), en el ámbito de *JavaScript*, pero extendido a múltiples lenguajes en la actualidad.

Es sencillo de manejar por máquinas y humanos.

A efectos de *JavaScript* es un objeto, convertible en cadena de texto para intercambio con servidores.

JSON

Formato básico

Está construido a partir de dos estructuras básicas:

- Una colección de pares nombre/valor → Dependiendo del lenguaje se implementa como: objeto, registro, estructura, diccionario, tabla hash,…
- Una lista ordenada de valores → En la mayor parte de los lenguajes se implementa como array, vector o lista.

JSON

Formato básico

En JavaScript las dos estructuras son:

- Objetos

{cadena:valor, cadena:valor, …….}

- Arrays

[valor, valor, …..]

Siendo los valores cadenas, números, objetos, arrays, booleanos o null.

```
miObjeto= {asignatura: "DAW", centro: "ETSE", curso: 4}

miArray= ["Antonio", "José", "Pedro"]
```

JSON

JavaScript cuenta con la clase JSON y métodos para:

- Transformar objetos en cadenas JSON

```
var miCadenaJSON= JSON.stringify(miObj)
```

- Transformar cadenas JSON en objetos JavaScript

```
var miObjeto= JSON.parse(miCadenaJSON)
```

XML vs JSON

XML

```
<alumnos>
    <alumno>
        <nombre>Pedro</nombre>
        <apellido>Gómez<apellido>
    </alumno>
    <alumno>
        <nombre>Pablo</nombre>
        <apellido>Pérez<apellido>
    </alumno>
        …
</alumnos>
```

JSON

```
{"alumnos" : [
   {"nombre" : "Pedro", "apellido" : "Gómez"},
   {"nombre" : "Pablo", "apellido" : "Pérez"},
    …
]}
```

XML vs JSON

Similitudes

Ambos son:

- Autodescriptivos
- Jerárquicos
- Analizables *(Parseables)*
- Tratables por el método XMLHttpRequest de JavaScript

Diferencias

- XML tiene que ser analizado por un analizador (*parser)* XML. JSON puede utilizar funciones Javascript estándar
- JSON no usa etiquetas
- JSON es más rápido, tanto en lectura como en escritura
- JSON puede utilizar arrays