
1. Llenguatges de marques

Desenvolupament web en entorn servidor

Vicent Jordà

16/11/2020

Índex de continguts

1	Llenguatge de marques	3
2	Llenguatges de marques	3
2.1	Concepte de llenguatge de marques	3
2.2	Avantatges. Necessitat d'ús	3
2.3	Característiques comunes	4
2.4	Identificació d'àmbits d'aplicació	4
2.5	Classificació	5
3	XML	8
3.1	Introducció	8
3.2	Estructura i sintaxi	8
3.3	Normes de sintaxi bàsiques en XML	11
3.4	Atributs en XML	13
3.5	Declaració XML	15
3.6	Referència a altres entitats	16
3.7	Comentaris en XML	21
3.8	Seccions CDATA en XML	24
3.9	Ferramentes d'edició	25
3.10	Elaboració de documents XML ben formats	26
3.11	Utilització d'espais de noms en XML	26
3.12	Bibliografia, webgrafia i crèdits	29

1 Llenguatge de marques

2 Llenguatges de marques

2.1 Concepte de llenguatge de marques

Un llenguatge de marcat o llenguatge de marques és una forma de codificar un document que, juntament amb el text, incorpora etiquetes o marques que contenen informació addicional sobre l'estructura del text o la seva presentació.

El llenguatge de marques més estès és l'HTML (HyperText Markup Language, llenguatge de marcat d'hipertext), fonament del World Wide Web (xarxa de comunicació d'abast mundial).

Alguns exemples:

HTML 4.0

```
1 <h1>Anatidae</h1>
2 <p>
3   The family <i>Anatidae</i> includes ducks, geese, and swans,
4   but <em>not</em> the closely related screamers.
5 </p>
```

XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <note>
3   <to>Tove</to>
4   <from>Jani</from>
5   <heading>Reminder</heading>
6   <body>Don't forget me this weekend!</body>
7 </note>
```

2.2 Avantatges. Necessitat d'ús

Inicialment els llenguatges de marques es van enfocar a la generació de documents però gràcies als seus avantatges s'ha extès el seu ús envers la definició d'estructures de dades i la compartició d'informació.

Dels avantatges destaquem:

- La facilitat de creació i lectura.

- El compliment d'estàndards d'emmagatzematge definits i públics.
- La incorporació de metadades.
- La definició de l'estructura de les dades.

2.3 Característiques comunes

Els llenguatges de marques han destacat per una sèrie de característiques que els han convertit en els tipus de llenguatges més usats en la informàtica actual per emmagatzemar i representar les dades. Entre les característiques més interessants que ofereixen els llenguatges de marques hi ha:

- Que es basen en el text pla.
- Que permeten fer servir metadades.
- Que són fàcils d'interpretar i processar.
- Que són fàcils de crear i prou flexibles per representar dades molt diverses.

Les aplicacions d'Internet i molts dels programes d'ordinador que es fan servir habitualment fan servir d'alguna manera o altra algun llenguatge de marques.

2.4 Identificació d'àmbits d'aplicació

Quant als àmbits d'aplicació podem definir la següent classificació:

- Documents en general:
 - Llenguatges descriptius com YAML, EBML.
 - Llenguatges de presentació com RTF, Tex, HTML
 - Llenguatges lleugers com Markdown
- Tecnologies d'Internet:
 - HTML, XHTML, GladeXML, Atom, RSS, WSDL
- Llenguatges especialitzats:
 - SVG, XMPP, COLLADA

En [Lenguaje de marcado](#) trobareu més informació.

2.5 Classificació

És complicat fer una classificació dels llenguatges de marques que hi ha però generalment s'accepta que en tenim dos grans grups basant-nos en quin és l'objectiu bàsic del llenguatge de marques:

- **LLENGUATGES PROCEDIMENTALS I DE PRESENTACIÓ**, orientats a especificar com s'ha de representar la informació.
- **LLENGUATGES DESCRIPTIUS O SEMÀNTICS**: orientats a descriure l'estructura de les dades que conté. Aquesta és la classificació més acceptada però, com molt sovint passa en l'àmbit de la Informàtica, ens podem trobar llenguatges que tinguin aspectes dels dos grups i permetin tant definir la manera de presentar la informació com definir-ne l'estructura.

Procedimentals i de presentació

En aquests llenguatges el que es fa és indicar de quina manera s'ha de fer la presentació de les dades. Ja sigui per mitjà d'informació per al disseny (marcar negretes, títols, etc.) o de procediments que ha de fer el programari de representació. L'exemple més popular d'aquests llenguatges és l'HTML però n'hi ha molts més: TeX, Wikitext...

En aquests casos els documents ens poden servir per determinar de quina manera es mostrarà el document a qui el llija.

Latex: un llenguatge de marcat procedural Per exemple:

```
1 \documentclass{article}
2 \usepackage{graphicx}
3
4 \begin{document}
5
6 \title{Introduction to \LaTeX{}}
7 \author{Author's Name}
8
9 \maketitle
10
11 \begin{abstract}
12 The abstract text goes here.
13 \end{abstract}
14
15 \section{Introduction}
16 Here is the text of your introduction.
```

```
17
18 \begin{equation}
19   \label{simple_equation}
20   \alpha = \sqrt{\beta}
21 \end{equation}
22
23 \subsection{Subsection Heading Here}
24 Write your subsection text here.
25
26 \begin{figure}
27   \centering
28   \includegraphics[width=3.0in]{myfigure}
29   \caption{Simulation Results}
30   \label{simulationfigure}
31 \end{figure}
32
33 \section{Conclusion}
34 Write your conclusion here.
35
36 \end{document}
```

Descriptius o semàntics

En aquests llenguatges es descriu quina estructura lògica té el document ignorant de quina manera serà representada en els programes. Només es posen les marques amb l'objectiu de definir les parts que donen estructura al document. L'exemple més important és l'XML però n'hi ha algun altre que està tenint molt de suport, com per exemple JSON. En el document següent tenim un exemple d'un fitxer de marques que dona informació sobre persones:

```
1 <alumnes>
2   <persona>
3     <nom>Pere</nom>
4     <cognom>Puig</cognom>
5   </persona>
6   <persona>
7     <nom>Manel</nom>
8     <cognom>Garcia</cognom>
9   </persona>
10 </alumnes>
```

Es pot veure clarament que la informació de les marques en el document estableix quin és el contingut de les dades: una llista d'alumnes. Amb un simple cop d'ull resulta fàcil determinar que Pere i Manel són noms i que Puig i Garcia són cognoms. Però per mitjà de la jerarquia de les dades es pot inferir que

Pere Puig i Manel Garcia són alumnes ja que tant nom com cognom estan englobats dins de la marca alumnes.

Aquest document mostra quina és l'estructura de les dades que conté i a més aquesta també es pot descobrir tot interpretant les etiquetes el seu contingut semàntic. A partir dels coneixements que es tinguin es dedueix que Pere és el nom d'una persona que és un alumne.

En els darrers anys, s'han desenvolupat diversos llenguatges de marques senzills i poc estandarditzats per permetre als autors crear text formatat a través de navegadors web, com els que s'utilitzen en wikis i en fòrums web. De vegades s'anomenen llenguatges de marcatge lleugers. Markdown i el llenguatge de marques utilitzat per la Viquipèdia en són exemples.

Exercici pràctic

Activitat 1. Exercicis

1. Compara les diverses possibilitats d'exportació, a l'hora de guardar un document en Libre-Office Writer.
2. Escriu en un fitxer de text pla el següent:

```
1 <h1> encapçalat de nivell 1 </h1>
2 <h3> encapçalat de nivell 3 </h3>
```

Obri este fitxer des del navegador. Com es veu? Canvia-li la extensió del fitxer per html. Canvia la visualització?

3. Busca en Internet una classificació de llenguatges de marques. Indica 3 exemples de cada tipus indicant el seu ús principal.
4. Entra en openclipart.org, cerca "8 ball", descarrega la imatge més senzilla que veges. Obre-la amb l'editor de text. Pots intuir l'estructura? Mira de canviar el color de la bola des de l'editor (pista: els colors tenen el format: #000000).

3 XML

3.1 Introducció

XML (*eXtensible Markup Language*, Llenguatge de Marcat extensible) és un llenguatge desenvolupat per W3C (World Wide Web Consortium) que està basat en SGML (*Standard Generalized Markup Language*, Llenguatge de Marcat Generalitzat Estàndard).

Informació

El World Web Consortium

El *World Wide Web Consortium* (W3C) és la principal organització internacional d'estàndards per a la *World Wide Web*. Fundat el 1994 i actualment dirigit per Tim Berners-Lee, el consorci està format per organitzacions membres que mantenen personal a temps complet treballant junts en el desenvolupament d'estàndards per a la *World Wide Web*. El W3C també es dedica a l'educació i divulgació, desenvolupa programari i serveix de fòrum obert per a la discussió sobre la web.

[Especificació de XML 1.0](#)

XML és un llenguatge utilitzat per a l'emmagatzematge i intercanvi de dades estructurades entre diferents plataformes.

XML és un metallenguatge, és a dir, pot ser emprat per a definir altres llenguatges, anomenats *dialectes XML*. Per exemple, alguns llenguatges basats en XML són:

- GML (*Geography Markup Language*, Llenguatge de Marcat Geogràfic).
- MathML (*Mathematical Markup Language*, Llenguatge de Marcat Matemàtic).
- RSS (*Really Simple Syndication*, Sindicació Realment Simple).
- SVG (*Scalable Vector Graphics*, Gràfics vectorials escalables).
- XHTML (*eXtensible HyperText Markup Language*, Llenguatge de Marcat d'Hipertext extensible).

3.2 Estructura i sintaxi

Els documents XML estan formats per text pla (sense format) i contenen marques (etiquetes) definides pel desenvolupador. CADA ETIQUETA REPRESENTA UN ELEMENT. Aquestes marques, és recomanable que siguin el més descriptives possible i, per escriure-les, s'utilitzen els caràcters menor que "<", major que ">" i barra inclinada (*slash*) "/".

Si en un document XML es vol guardar el nom Elsa, es pot escriure:

```
1 <nom> Elsa </nom>
```

La sintaxi utilitzada en l'exemple és la bàsica per escriure un element en XML:

```
1 <etiqueta> valor </etiqueta>
```

Cal observar que, entre l'etiqueta d'inici (<nom>) i l'etiqueta de fi (</ nom>) s'ha escrit la dada (valor) que es vol emmagatzemar. En aquest cas Elsa.

Elements buits

En un document XML, un element pot no contenir cap valor. En aquest cas cal escriure:

```
1 <etiqueta> </ etiqueta>
```

Es pot expressar el mateix escrivint:

```
1 <etiqueta />
```

Per escriure l'element nom buit, es pot escriure:

```
1 <nom> </nom>
```

O també:

```
1 <nom />
```

Relacions pare-fill entre elements

D'altra banda, un element (pare) pot contenir a un altre o altres elements (fills):

```
1 <persona>
2   <nom> Elsa </nom>
3   <dona />
4   <data-de-naixement>
5     <dia> 18 </dia>
6     <mes> 6 </mes>
7     <any> 1996 </any>
8   </data-de-naixement>
9   <ciutat> Pamplona </ciutat>
10 </persona>
```

En aquest exemple, l'element "persona" conté quatre elements (fills): "nom", "dona", "data de naixement" i "ciutat". I l'element "data de naixement" conté altres tres elements (fills): "dia", "mes" i "any".

Vegeu que, de tots els elements que apareixen en aquest exemple, només l'element "dona" està buit.

Element arrel d'un document XML

Tot document XML ha de tenir un únic element arrel (pare) del que descendiran tots els altres. En aquest cas, l'element arrel és "persona". Gràficament, l'estructura d'elements d'aquest document es pot representar com es mostra a continuació:



Figure 1: Estructura d'elements d'un document XML persona - Exemple d'el tutorial de XML de {Abrirllave.com

D'aquesta manera, l'estructura de qualsevol document XML es pot representar com un arbre invertit d'elements. Es diu que els elements són els que donen estructura semàntica a el document.

Elements amb contingut mixt Un element pot contenir contingut mixt, és a dir, text i altres elements:

```
1 <persona>
2   <nom>Elsa</ nom> viu a <ciutat>Pamplona</ ciutat>.
3 </ persona>
```

En aquest exemple, l'element “persona” conté els elements “nom” i “ciutat”, a més dels textos “viu en” i “.”.

3.3 Normes de sintaxi bàsiques en XML

En un document XML, tots els noms dels elements són *case sensitive*, és a dir, sensibles a lletres minúscules i majúscules, havent de complir les següents normes:

Poden contenir lletres minúscules, lletres majúscules, números, punts “.”, Guions mitjans “-” i guions baixos “_”.

Així mateix, poden contenir el caràcter dos punts “:”. No obstant això, el seu ús es reserva per quan es defineixin espais de noms.

El primer caràcter ha de ser una lletra o un guió baix “_”.

D'altra banda, cal tenir en compte que darrere del nom d'una etiqueta es permet escriure un espai en blanc o un salt de línia. Per exemple, sintàcticament és correcte escriure:

```
1 <ciutat> Pamplona </ ciutat>
2 >
```

Ara bé, no hi pot haver un salt de línia o un espai en blanc abans del nom d'una etiqueta:

```
1 <
2 ciutat> Pamplona </ ciutat>
```

Els següents elements no estan escrits correctament per incomplir alguna regla de sintaxi:

```
1 <ciudad>Pamplona</ciudad>
2 <día>18</dia>
3 <mes>6<mes/>
4 <ciudad>Pamplona</finciudad>
5 <_rojo>
6 <2colores>Rojo y Naranja</2colores>
```

```
7 < Aficiones >Cine, Bailar, Nadar</ Aficiones >
8 <persona><nombre>Elsa</persona></nombre>
9 <color favorito>azul</color favorito>
```

S'hauria d'escriure:

```
1 <ciudad>Pamplona</ciudad>
2 <día>18</día>
3 <mes>6</mes>
4 <ciudad>Pamplona</ciudad>
5 <_rojo/>
6 <colores2>Rojo y Naranja</colores2>
7 <Aficiones >Cine, Bailar, Nadar</Aficiones >
8 <persona><nombre>Elsa</nombre></persona>
9 <color.favorito>azul</color.favorito>
10 <color-favorito>azul</color-favorito>
11 <color_favorito>azul</color_favorito>
```

Les lletres no angleses (á, Á, ñ, Ñ ...) estan permeses. No obstant això, és recomanable no utilitzar-les per reduir possibles incompatibilitats amb programes que puguin no reconèixer-les.

Pel que fa al caràcter guió "-" i a punt ".", Encara que també estan permesos per nomenar etiquetes, igualment s'aconsella evitar el seu ús; el guió perquè podria confondre amb el signe menys, i el punt perquè, per exemple a l'escriure color.favorito, podria interpretar-se que favorito és una propietat de l'objecte color.

Exercici pràctic

Activitat 2. Errors de sintaxi

Revisa el següent document XML, detecta els errors i reescriu-lo correctament

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <frutas>
3 </frutas>
4 < frutas >
5     < fruta >
6         < nombre >cereza< nombre \>
7     < fruta \>
8     < fruta >
9         < nombre >naranja< nombre \>
10    < fruta \>
11 < frutas \>
```

3.4 Atributs en XML

Els elements d'un document XML poden tenir atributs definits en l'etiqueta d'inici. Un atribut serveix per proporcionar informació extra sobre l'element que el conté.

Donats els següents dades d'un producte:

- Codi: G45
- Nom: Barret de llana
- Color: negre
- Preu: 12.56

La seva representació en un document XML podria ser, per exemple:

```
1 <producte cod = "G45">
2   <nom color = "negre" preu = "12.56"> Barret de llana </ nom>
3 </ producte>
```

En aquest exemple s'han escrit tres atributs: codi, color i preu. Cal observar que, els seus valors ("G45", "negre" i "12.56") s'han escrit entre cometes dobles ("). Tanmateix, també poden anar entre cometes simples (').

Si, per exemple, l'atribut `codi` es volgués representar com un element, es podria escriure:

```
1 <producte>
2   <codi> G45 </ codi>
3   <nom color="negre" preu="12.56">Barret de llana</ nom>
4 </ producte>
```

Com es pot apreciar, ara el valor del codi no s'ha escrit entre cometes dobles.

Exercici pràctic

Elements i atributs

Un element és un component lògic d'un document XML. Els elements solen tenir entitat pròpia. El contingut d'un element és tot el que es troba entre les etiquetes d'obertura i tancament, inclús si contenen altres elements (fills).

Per contra, els atributs solen representar propietats o característiques dels elements.

Normes de sintaxi

Els noms dels atributs han de complir les mateixes normes de sintaxi que els noms dels elements. A més, tots els atributs d'un element han de ser únics. Per exemple, és incorrecte escriure:

```
1 <dades x="3" x="4" i="5" />
```

No obstant això, sí que és correcte escriure:

```
1 <dades x="3" X="4" i="5" />
```

Els atributs continguts en un element, com en aquest cas x, X i y, han de separar amb espais en blanc, i no és significatiu el seu ordre.

Exercici pràctic**Activitat 3. Creació de documents XML****Ciutats**

Nom	País	Continent
Nueva Delhi	India	Àsia
Lisboa	Portugal	Europa
El Cairo	Egipto	Àfrica

Nota: el continent es representarà mitjançant un atribut.

Fets històrics

Descripció de cada fet	Data		
	Dia	Mes	Any
IBM dona a conèixer el PC.	12	8	1981
Es funda Google	4	9	1998
Es funda Facebook.	4	2	2004

Figure 2: Fets històrics

Nota: la descripció de cada fet cal representar-la mitjançant un atribut, la resta d'informació no.

3.5 Declaració XML

La declaració XML que es pot escriure al principi d'un document XML comença amb els caràcters "<?" i acaba amb ">".

Versió i codificació

Un document XML podria contenir la següent declaració XML:

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

En aquesta declaració XML, està indicant que 1.0 és la versió de XML utilitzada en el document i UTF-8 (8-bit *Unicode Transformation Format*, Formato de Transformación Unicode de 8 bits) és la codificació de caràcters empleada.

En un document XML no és obligatori que aparega la declaració XML. Ara bé, si la inclou, ha d'aparèixer en la primera línia del document, i el caràcter "<" ha de ser el primer de dita línia, és dir, abans no poden aparèixer espais en blanc.

Atribut *standalone*

En una declaració XML, a més de la versió i la codificació dels atributs, també es pot escriure l'atribut *standalone*, que pot prendre dos valors ("yes" o "no"):

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

En escriure *standalone="yes"* s'està indicant que el document és independent d'altres, com per exemple d'una DTD (*Document Type Definition*, Definició de Tipo de Documento) externa (o vorem més endavant). En cas contrari, significarà que el document no és independent.

En un document XML, escriure la declaració XML és opcional. Però, si s'escriu, l'atribut *version* és obligatori. No obstant, els atributs *encoding* i *standalone* són opcionals i, per defecte, els seus valors són "UTF-8" i "no", respectivament.

Per una altra part, quan s'escriu l'atribut *encoding*, sempre haurà d'aparèixer després de la versió. I, l'atribut *standalone*, sempre que existisca, haurà d'estar en l'últim lloc.

3.6 Referència a altres entitats

En XML alguns caràcters que són especials pel seu significat (veure taula) i, per escriure'ls en un document XML, es poden utilitzar les referències a entitats mostrades en la següent taula:

Caràcter	Entidad	Referencia a entidad
< (menor que)	lt (<i>less than</i>)	<
> (mayor que)	gt (<i>greater than</i>)	>
” (comilla doble)	quot (<i>quotation mark</i>)	"
' (comilla simple)	apos (<i>apostrophe</i>)	'
& (ampersand)	amp (<i>ampersand</i>)	&

Donat l'arxiu “entidades.xml”:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <entidades>
3   <menor_que>&lt;</menor_que>
4   <mayor_que>&gt;</mayor_que>
5   <comilla_doble>&quot;</comilla_doble>
6   <comilla_simple>&apos;</comilla_simple>
7   <ampersand>&amp;</ampersand>
8 </entidades>
```

En obrir-lo en Google Chrome es podrà visualitzar:

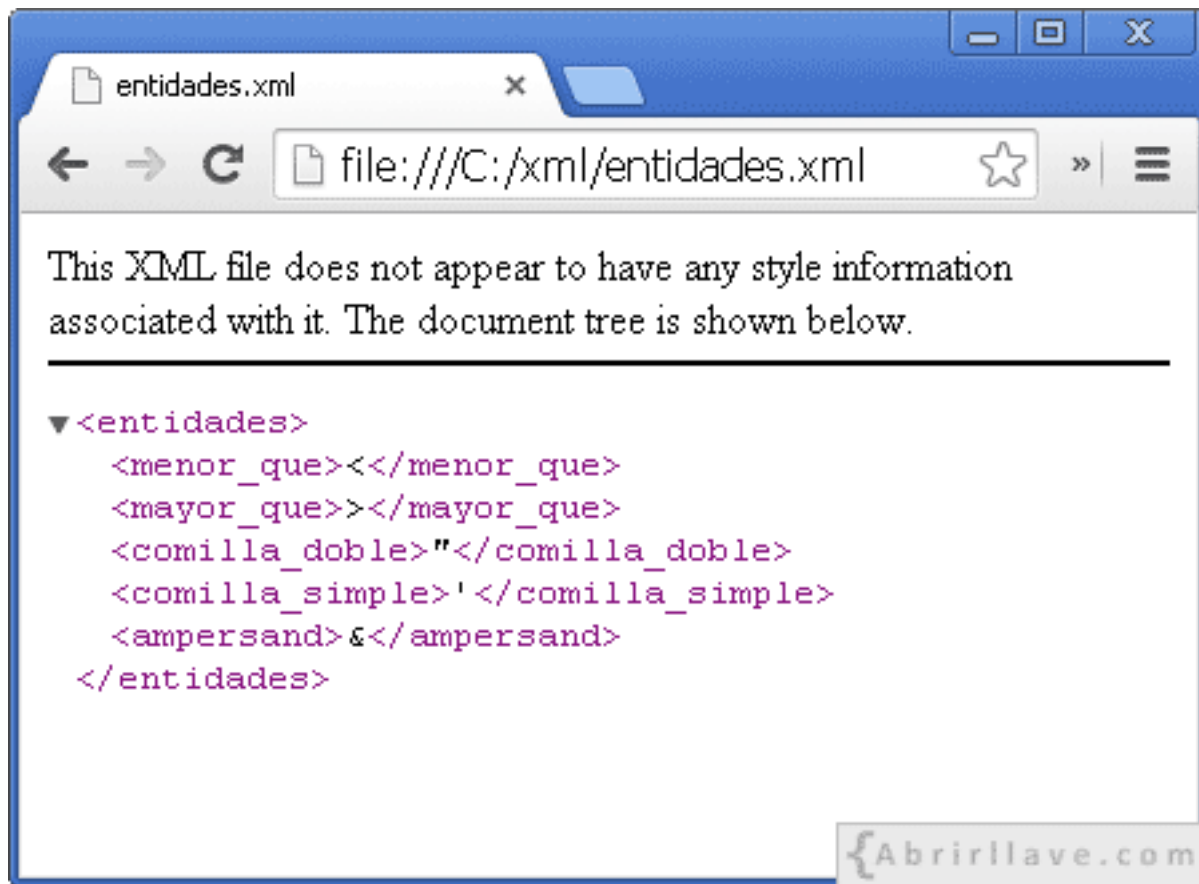


Figure 3: Visualització de l'arxiu entidades.xml a Google Chrome - Exemple d'el tutorial de XML de Abrirllave.com

Al navegador web, es pot veure que on s'han escrit les referències a entitats en el document XML (per exemple), es mostren els caràcters corresponents (per exemple <).

Caràcters problemàtics en XML: menor que (<) i ampersand (&)

En un document XML, el caràcter "<" és problemàtic perquè indica el començament d'una etiqueta. Per tant, en comptes d'escriure, per exemple:

```
1 <condicion>a<b</condicion>
```

Caldria utilitzar la referència a entitat escrivint:

```
1 <condicion>a<lt;b</condicion>
```

El caràcter ">" sí que pot utilitzar-se en el text contingut en un element, i no és incorrecte escriure, per exemple:

```
1 <condicion>a>b</condicion>
```

Ara bé, es recomana fer ús de la seva referència a entitat (>)

En un document XML, el caràcter *ampersand* també és problemàtic, ja que s'utilitza per indicar el començament d'una referència a entitat. Per exemple, no és correcte escriure:

```
1 <condicion>a==1 && b==2</condicion>
```

En el seu lloc s'ha d'escriure el següent:

```
1 <condicion>a==1 &amp;&amp; b==2</condicion>
```

Ús de la cometa doble (") i de la cometa simple (') en atributs

Si el valor d'un atribut s'escriu entre cometes dobles ("), aquest valor no podrà contenir aquest caràcter. Per exemple, no és correcte escriure:

```
1 <dato character="comilla doble(">"/>
```

Per a això, cal utilitzar la referència a entitat com es mostra a continuació:

```
1 <dato character="comilla doble(&quot;>"/>
```

De la mateixa manera passa amb la cometa simple ('), sent incorrecte escriure, per exemple:

```
1 <dato character='comilla simple(>'>"/>
```

Pel que, en aquest cas, caldria usar com es mostra tot seguit:

```
1 <dato character='comilla simple(&apos;>')>"/>
```

D'altra banda, els valors d'atributs escrits entre cometes dobles (") sí que poden contenir el caràcter cometa simple (') i al revés. Per exemple, és correcte escriure:

```
1 <dato character="comilla simple(>')>"/>
2 <dato character='comilla doble(">')>"/>
```

En aquests casos, *no és obligatori fer servir les referències a entitats, però sí recomanable*.

Referències de caràcters en XML

En un document XML es poden escriure referències de caràcters Unicode amb els símbols & #, seguits de la valor decimal o hexadecimal de l'caràcter Unicode que es vulgui representar i, finalment, afegint el caràcter punt i coma “;”.

Representació del caràcter Euro (€) en XML

Donat el document XML “productos.xml”:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <productos>
3   <nombre precio="12.56&#8364;">Gorro de lana</nombre>
4   <nombre precio="16.99&#x20AC;">Gorro polar</nombre>
5 </productos>
```

En visualitzar en un navegador web, es podrà veure el següent:

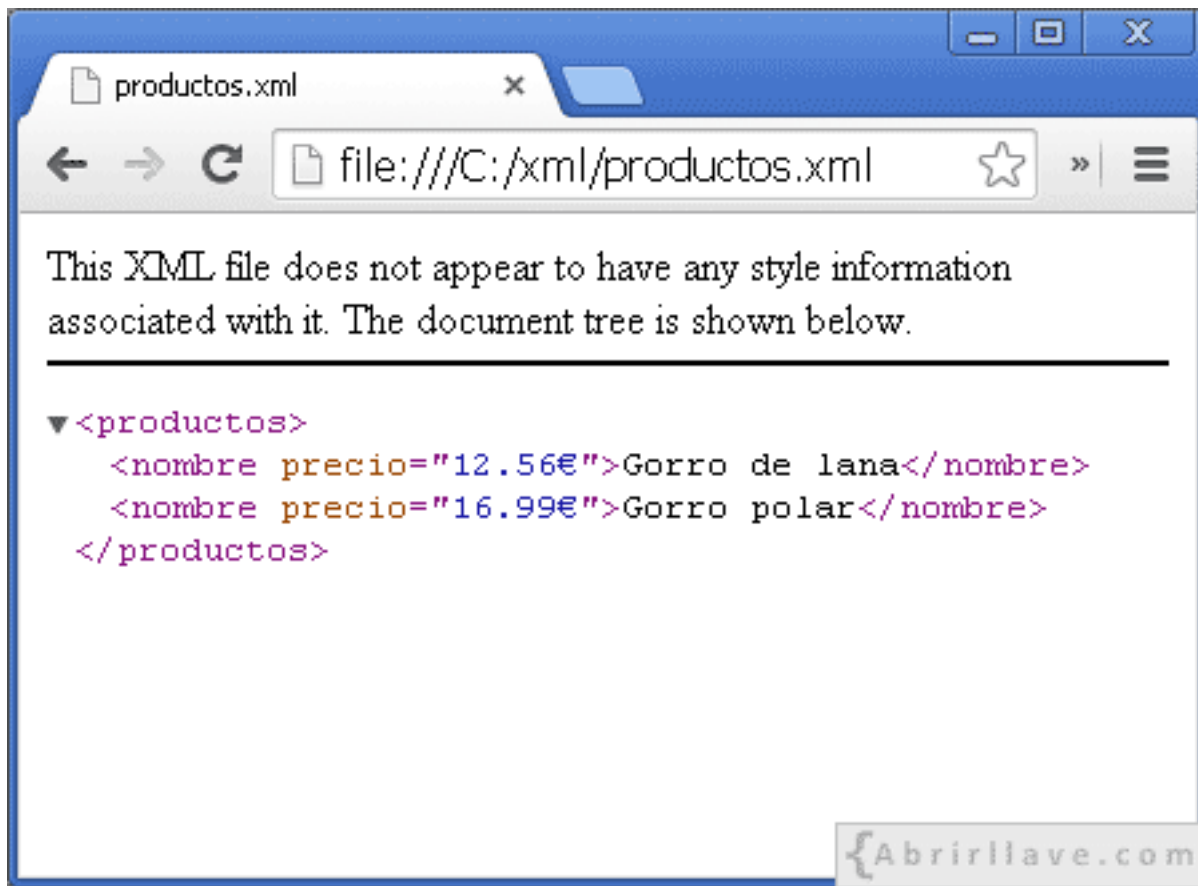


Figure 4: Visualització de l'arxiu productos.xml a Google Chrome - Exemple d'el tutorial de XML de {Abrirllave.com}

Cal observar que, en aquest cas, per representar el símbol de l'Euro (€), la primera vegada s'ha utilitzat el seu valor decimal (€) en Unicode i, la segona vegada, el seu valor hexadecimal (€).

3.7 Comentarís en XML

Per escriure comentaris en un document XML, aquests s'han d'escriure entre els caràcters “<!--” i “-->”. Per exemple:

```
1 <!-- Això és un comentari escrit en un document XML -->
```

Donat el fitxer XML “letras.xml”:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--Ejemplo uso de comentarios.-->
3 <a>
4     <b>
5         <c cantidad="4">cccc</c>
6         <d cantidad="2">dd</d>
7     </b>
8     <e>
9         <f cantidad="8">ffffffff</f>
10        <!--g puede aparecer varias veces.-->
11        <g cantidad="5">ggggg</g>
12        <g cantidad="2">gg</g>
13    </e>
14 </a>
```

En un navegador es veurà:



Figure 5: Visualització de l'arxiu letras.xml a Google Chrome - Exemple d'el tutorial de XML de {Abrirllave.com}

En un document XML, no es poden escriure comentaris dins de les etiquetes. Per exemple, no és correcte escriure:

```
1 <element <!-- element buit -> />
```

D'altra banda, cal tenir en compte que en els comentaris d'un document XML no està permès utilitzar dos guions seguits:

```
1 <!-- dos guions seguits -- en un comentari dona error ->
```

De manera que, no és possible niar comentaris en un document XML.

3.8 Seccions CDATA en XML

Un document XML pot contenir seccions CDATA (*Character DATA*) per escriure text que no es desitja que sigui analitzat. Per exemple, això pot ser útil quan es vol escriure text que continga algun dels caràcters problemàtics: menor que < o ampersand &.

En un document XML, per incloure una secció CDATA, s'escriu començant amb la cadena de caràcters "<![CDATA [" i acabant amb els caràcters "]]>".

Una secció CDATA pot contenir, per exemple, el codi font d'un programa escrit en llenguatge C:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ejemplo_CDATA>
3 <![CDATA[
4 #include <stdio.h>
5 int main()
6 {
7     float nota;
8     printf( "\n   Introduzca nota (real): " );
9     scanf( "%f", &nota );
10    if ( 5 <= nota )
11        printf( "\n   APROBADO" );
12    return 0;
13 }
14 ]]>
15 </ejemplo_CDATA>
```

En un navegador web es visualitzarà cosa semblant a:

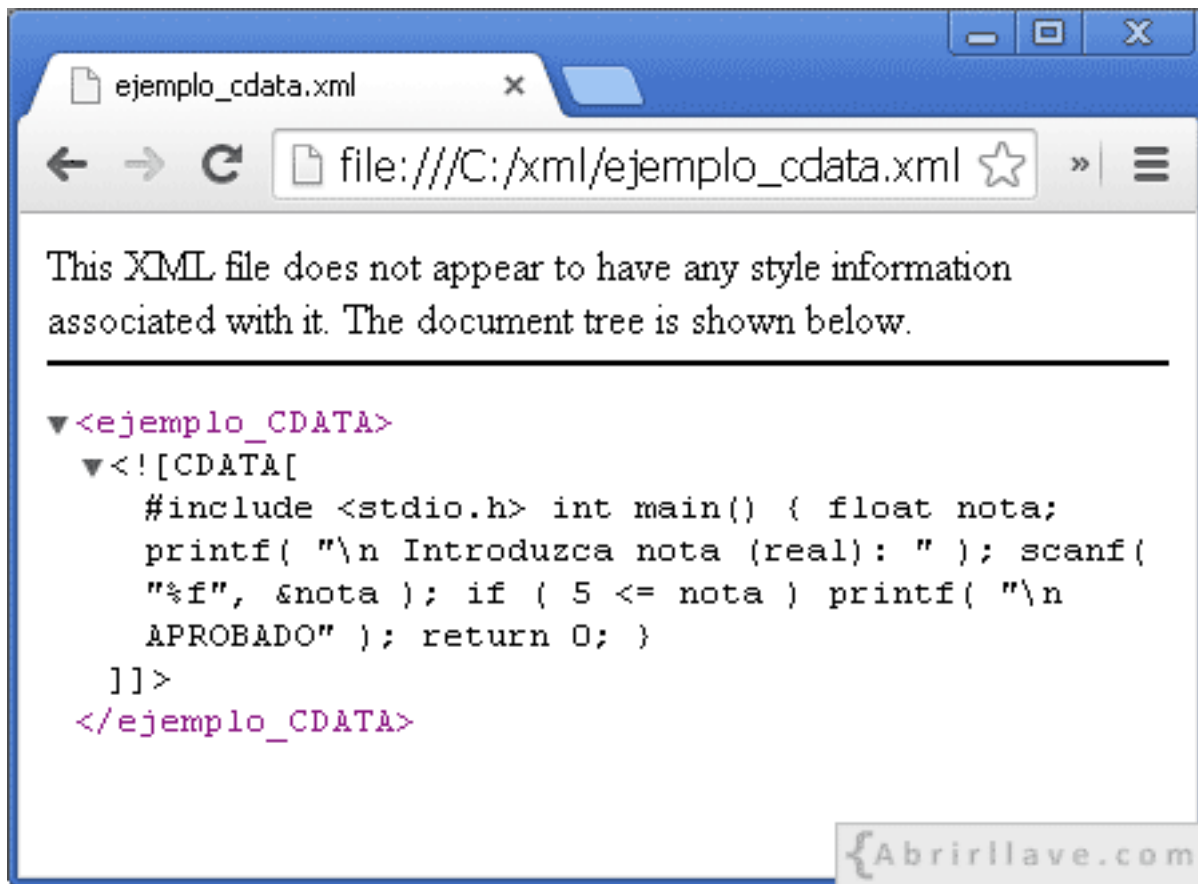


Figure 6: Visualització de l'arxiu ejemplo_cdata.xml a Google Chrome - Exemple d'el tutorial de XML de {Abrirllave.com}

Dins d'una secció CDATA no es pot escriure la cadena “]]>”. En conseqüència, no es poden niar seccions CDATA.

D'altra banda, no està permès escriure espais en blanc o salts de línia en les cadenes d'inici “<![CDATA[” o fi “]]>” d'una secció CDATA

3.9 Ferramentes d'edició

Per a editar documents XML és suficient en disposar d'un editor de text pla, com Pluma o GEdit, però també podem utilitzar alguns editors específics com XML Copy Editor o un IDE amb algun connector com Visual Studio Code amb XML Tools.

3.10 Elaboració de documents XML ben formats

Es diu que un document XML està ben format (well-formed document) quan no té errors de sintaxi. Això inclou els següents aspectes:

- Els noms dels elements i els seus atributs han d'estar escrits correctament.
- Els valors dels atributs han d'estar escrits entre cometes dobles o simples.
- Els atributs d'un element s'han de separar amb espais en blanc.
- S'han d'utilitzar referències a entitats on siga necessari.
- Hi ha d'haver un únic element arrel.
- Tot element ha de tenir un element pare, excepte l'element arrel.
- Tots els elements han de tenir una etiqueta d'obertura i una altra de tancament.
- Les etiquetes han d'estar correctament niades.
- La declaració XML ha d'estar en la primera línia escrita correctament.
- Les seccions `CDATA` i els comentaris han d'estar correctament escrits.

Exercici pràctic

Activitat 4: Els mòduls de DAW

Crea un document XML que emmagatzeme els mòduls del cicle de DAW indicant les hores de cada mòdul i el curs en què s'imparteix.

Exercici pràctic

Activitat 5: Documents XML ben format

Enunciat de l'activitat

3.11 Utilització d'espais de noms en XML

Els espais de noms XML són un mecanisme per assegurar que els elements i els atributs d'un document XML tinguin noms únics. Estan definits en una recomanació del W3C. El problema que resolen és l'ambigüitat que sorgeix quan un document XML conté noms d'elements o d'atributs provinents de diversos vocabularis i en resulten diversos elements o atributs homònims (amb el mateix nom): si a cada vocabulari se li dóna un espai de noms diferent, es resol l'ambigüitat.

Per exemple:

```
1 <carta>
2   <palo>Corazones</palo>
3   <numero>7</numero>
4 </carta>
```

```
1 <carta>
2   <carnes>
3     <filete_de_tenera precio="12.95"/>
4     <solomillo_a_la_pimienta precio="13.60"/>
5   </carnes>
6   <pescados>
7     <lenguado_al_horno precio="16.20"/>
8     <merluza_en_salsa_verde precio="15.85"/>
9   </pescados>
10 </carta>
```

De forma que, si se inclouen els elements `<carta>` en un document XML, s'origina un conflicte de noms. Per a resoldre'l, es poden utilitzar espais de noms (XML Namespaces). Per exemple, escrivint:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <e1:ejemplo xmlns:e1="http://www.abrirllave.com/ejemplo1"
3   xmlns:e2="http://www.abrirllave.com/ejemplo2">
4
5   <e1:carta>
6     <e1:palo>Corazones</e1:palo>
7     <e1:numero>7</e1:numero>
8   </e1:carta>
9
10  <e2:carta>
11    <e2:carnes>
12      <e2:filete_de_tenera precio="12.95"/>
13      <e2:solomillo_a_la_pimienta precio="13.60"/>
14    </e2:carnes>
15    <e2:pescados>
16      <e2:lenguado_al_horno precio="16.20"/>
17      <e2:merluza_en_salsa_verde precio="15.85"/>
18    </e2:pescados>
19  </e2:carta>
20 </e1:ejemplo>
```

Per a definir un espai de noms s'utilitza la següent sintaxi:

```
1 xmlns:prefijo="URI"
```

En l'exemple, observa que, `xmlns` és un atribut que s'ha utilitzat en l'etiqueta d'inici de l'element

<ejemplo> i, en aquest cas, s'han definit dos espais de noms que fan referència a los siguientes URI (*Uniform Resource Identifier*, Identificador Uniforme de Recurs):

- `http://www.abrirllave.com/ejemplo1`
- `http://www.abrirllave.com/ejemplo2`

Los prefixes definits són e1 i e2, respectivament. S'han afegit els prefixes a les etiquetes que apareixen en el document: <e1:carta>, <e2:carta>, <e1:palo>, etc.

Definició d'un espai de noms per defecte

D'altra banda, es pot definir un espai de noms per defecte mitjançant la següent sintaxi:

```
1 xmlns = "URI"
```

D'aquesta manera, tant l'element on s'ha definit l'espai de noms, com tots els seus successors (fills, fills de fills, etc.), han de pertànyer a aquest espai de noms. Per exemple:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ejemplo xmlns="http://www.abrirllave.com/ejemplo1">
3
4     <carta>
5         <palo>Corazones</palo>
6         <numero>7</numero>
7     </carta>
8
9 </ejemplo>
```

En el següent exemple, inicialment es defineix un espai de noms per defecte per a l'element <ejemplo> i els continguts en ell. Ara bé, posteriorment, es defineix un segon espai de noms, que per defecte afecta el segon element que apareix en el document i als seus successors: <carnses>, <pescados>, <filete_de_tenera> ...

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ejemplo xmlns="http://www.abrirllave.com/ejemplo1">
3
4     <carta>
5         <palo>Corazones</palo>
6         <numero>7</numero>
7     </carta>
8
9     <carta xmlns="http://www.abrirllave.com/ejemplo2">
```

```

10     <carnes>
11         <filete_de_tenera precio="12.95"/>
12         <solomillo_a_la_pimienta precio="13.60"/>
13     </carnes>
14     <pescados>
15         <lenguado_al_horno precio="16.20"/>
16         <merluza_en_salsa_verde precio="15.85"/>
17     </pescados>
18 </carta>
19
20 </ejemplo>

```

En un document XML, per indicar que determinats elements -o tots- no pertanyen a cap espai de noms, s'escriu l'atribut `xmlns` buit, és a dir, `xmlns = ""`.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ejemplo xmlns="http://www.abrirllave.com/ejemplo1">
3
4     <carta>
5         <palo>Corazones</palo>
6         <numero>7</numero>
7     </carta>
8
9     <carta xmlns="http://www.abrirllave.com/ejemplo2">
10         <carnes>
11             <filete_de_tenera precio="12.95"/>
12             <solomillo_a_la_pimienta precio="13.60"/>
13         </carnes>
14         <pescados xmlns="">
15             <lenguado_al_horno precio="16.20"/>
16             <merluza_en_salsa_verde precio="15.85"/>
17         </pescados>
18     </carta>
19
20 </ejemplo>

```

En aquest cas, l'element i els seus fills, no pertanyen a cap espai de noms.

3.12 Bibliografia, webgrafia i crèdits

- Wikipedia contributors. (2020, September 13). Markup language. In *Wikipedia, The Free Encyclopedia*. Retrieved 15:51, September 15, 2020, from https://en.wikipedia.org/w/index.php?title=Markup_language&oldid=978142210

- Carlos Pes. (Febrer de 2017). *Lenguajes de Marcas y Sistemas de Gestión de Información (LMSGI)* disponible en [Tutorial de LMSGI](#)
- Bartolomé Sintés Marco. (Juny de 2020). *XML: Lenguaje de marcas extensible*, disponible en <https://www.mclibre.org/consultar/xml/>