

# **XML**

---

eXtensible Markup Language

# Necessitat del XML

- Les aplicacions informàtiques fan servir una gran quantitat de dades, que han de ser consultades, intercanviades, manipulades etc.
- Cada aplicació desa les dades en un format propietari, de tal manera que per poder fer-los servir has de tenir el mateix programa.
- XML: **és una forma estàndar d'intercanviar dades**. Per exemple Word (doc a docx), Excel (xls axlsx) ja fan servir XML per desar les dades.

# Característiques

- Document de text pla (editors bàsics)
- Llenguatge de marques (estructura dades)
- Dissenyat pel transport de dades, no per mostrar-los (format amb CSS).
- Les etiquetes XML no definides, es defineixen per a cada aplicació.
- XML és autodescriptiu.
- Desenvolupat per la W3C.
- Eina **independent del software i del hardware** (per desar les dades en text pla).
- Tecnologies addicionals que complementen XML: XLT, Xpath, Xquery, Xforms...

# Metallenguatge

- XML:= és una forma de crear llenguatges.
- Alguns llenguatges basats en XML:
  - **GML** (*Geography Markup Language, Lenguaje de Mercado Geográfico*).
  - **MathML** (*Mathematical Markup Language, Lenguaje de Mercado Matemático*).
  - **RSS** (*Really Simple Syndication, Sindicación Realmente Simple*).
  - **SVG** (*Scalable Vector Graphics, Gráficos Vectoriales Escalables*).
  - **XHTML** (*eXtensible HyperText Markup Language, Lenguaje de Mercado de Hipertexto eXtensible*).

```
<?xml version="1.0" encoding="UTF-8"?>
<persona profesion="cantante">
  <nombre>Elsa</nombre>
  <mujer/>
  <fecha_de_nacimiento>
    <dia>18</dia>
    <mes>6</mes>
    <año>1996</año>
  </fecha_de_nacimiento>
  <ciudad>Pamplona</ciudad>
</persona>
```

És només informació entorn etiquetes

**Encapçalament.** Es defineix la versió. Opcional

I el cos

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

# XML vs HTML

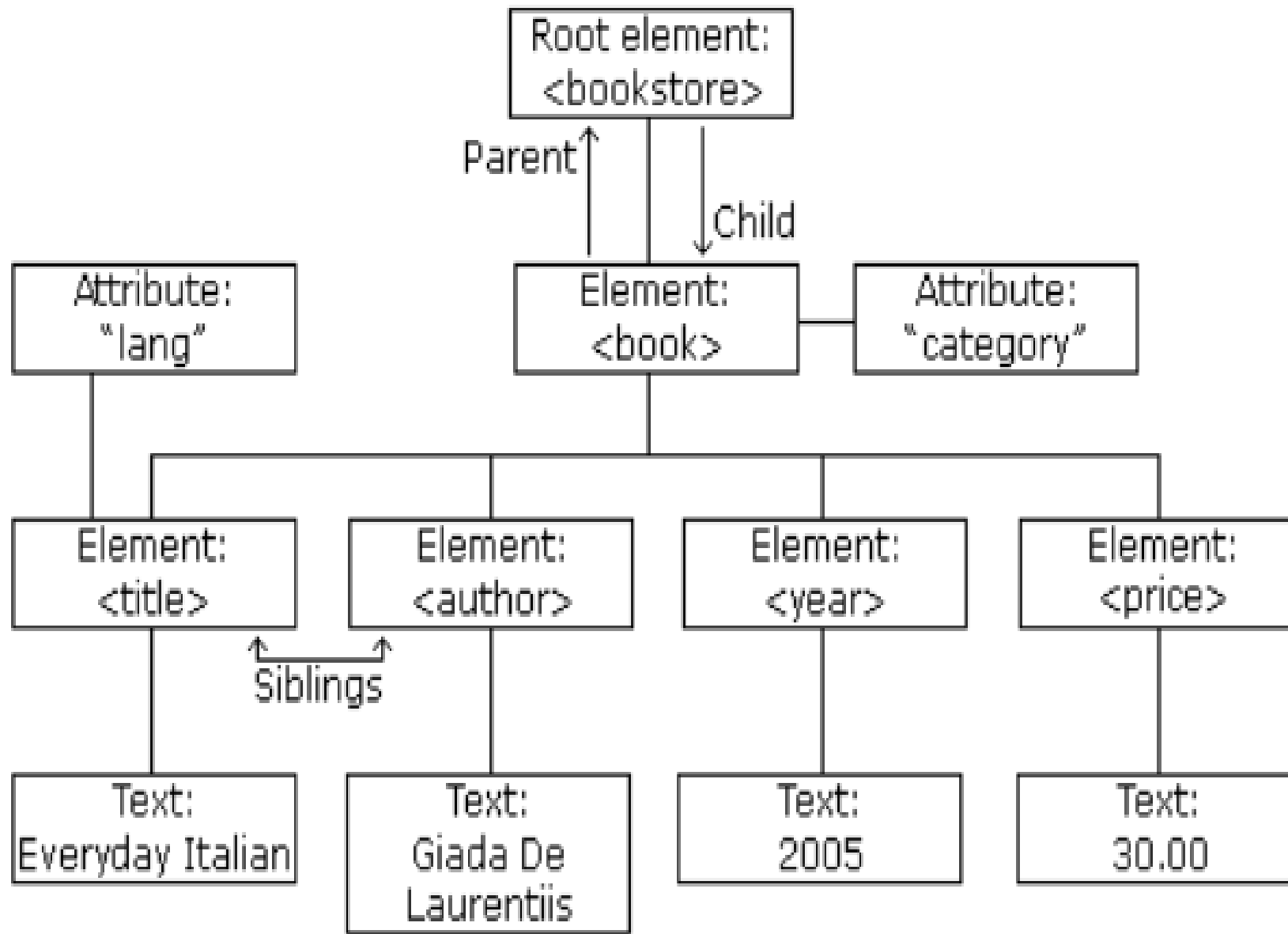
- **XML** va ser dissenyat per transportar dades, lo important són les dades .
- **HTML** va ser dissenyat per mostrar dades. Lo important és com es mostra les dades(estructura).
- **XML** no té les etiquetes predefinides, la persona autora és qui les defineix.
- **HTML** sí té les etiquetes predefinides.
- **XML** és extensible.

# Arbre XML

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

Pares, fills i germans

arbreXml.xml





# Sintaxi

## Objectiu: arxius XML ben estructurats

- Per desar un valor:

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

- Pot tenir un element buit:

```
<year></year>
```

```
<element></element>
```

---

```
<element />
```

# Sintaxi

- Els docs XML han de tenir un element arrel (<bookstore>).

- Declaració del XML:

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

- ✓ És opcional:
- ✓ Si es posa va a l'inici del document.
- ✓ S'especifica la versió XML emprada (abans de l'encoding).
- ✓ S'ha d'especificar la codificació emprada, la codificació de caràcters predeterminada per documents XML és la UTF-8 (igual que a HTML5, CSS, JS, PHP i SQL).

# Sintaxi

## Relació pare-fills

- L'element persona té quatre fills.
- I l'element fecha-de-nacimiento tres.

```
<persona>  
  <nombre>Elsa</nombre>  
  <mujer/>  
  <fecha-de-nacimiento>  
    <día>18</día>  
    <mes>6</mes>  
    <año>1996</año>  
  </fecha-de-nacimiento>  
  <ciudad>Pamplona</ciudad>  
</persona>
```

# Sintaxi

- Correctament anidats : `<b><i>This text is bold and italic</i></b>`
- En XML els elements també poden tenir atributs amb valors com l'HTML  
Aquests han d'estar citats. `<note date="12/11/2007">`
- Tots els elements XML han d'estar tancats (els tags).
- Elements amb continguts mixte:

```
<persona>  
  <nombre>Elsa</nombre> vive en <ciudad>Pamplona</ciudad>.  
</persona>
```

# Sintaxi

- XML és case sensitive distingeix entre majúscules i minúscules.

<carta> no és lo mateix que <Carta>.

```
<Price>29.99</price> incorrecte  
<Price>29,99</Price> correcte
```

- Espais en blanc:

```
<?xml version="1.0" encoding="UTF-8"?>  
<peliculas>  
  <pelicula>El discurso del rey</pelicula>  
  <pelicula>En      tierra      hostile</pelicula>  
  <pelicula>Una  
    mente  
  
maravillosa</pelicula>  
</peliculas>
```



```
<peliculas>  
  <pelicula>El discurso del rey</pelicula>  
  <pelicula>En tierra hostile</pelicula>  
  <pelicula>Una mente maravillosa</pelicula>  
</peliculas>
```

- Espais en blanc:

```
<?xml version="1.0" encoding="UTF-8"?>
<series>
  <serie numeros="2 4 6 8"/>
  <serie numeros="3
```

6

```
9
12 15"/>
</series>
```



```
▼<series>
  <serie numeros="2 4 6 8"/>
  <serie numeros="3 6 9 12 15"/>
</series>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<datos>
  <dato>1</dato>
  <dato>2</dato>
  <dato>3</dato>
</datos>
```



```
<datos>
  <dato>1</dato>
  <dato>2</dato>
  <dato>3</dato>
</datos>
```

datos.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<datos><dato>1</dato><dato>2</dato><dato>3</dato></datos>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<datos><dato>1</dato>  <dato>2</dato>
```

```
<dato>3</dato></datos>
```

# Atribut predefinit `xml:space` amb el valor `"preserve"`

```
<clasificacion xml:space="preserve">
1      Fernando Alonso      1:55.341
2      Lewis Hamilton       1:55.729
3      Sebastian Vettel     1:56.122
</clasificacion>
```

preserve.xml

Per indicar que els espais que apareixen en el contingut de l'element han de preservar-se.

No tots els programes el reconeixen.

Valors: preserve o default (el que es té per defecte).

# Sintaxi

- Referències a entitats, hi han cinc per XML:

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

entidades.xml

```
<year> any <2005</year>  
<year> any &lt;2005</year>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<entidades>  
  <menor_que>&lt;</menor_que>  
  <mayor_que>&gt;</mayor_que>  
  <comilla_doble>&quot;</comilla_doble>  
  <comilla_simple>&apos;</comilla_simple>  
  <ampersand>&amp;</ampersand>  
</entidades>
```



## Incorrecto

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

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

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

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

## Correcto

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

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

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

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

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

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

# Caràcters Unicode (ex:€)

```
<?xml version="1.0" encoding="UTF-8"?>
<productos>
  <nombre precio="12.56&#8364;">Gorro de lana</nombre>
  <!-- valor decimal-->
  <nombre precio="16.99&#x20AC;">Gorro polar</nombre>
  <!-- valor hexadecimal-->
</productos>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<productos>
  <nombre precio="12.56€">Gorro de lana</nombre>
  <nombre precio="16.99€">Gorro polar</nombre>
</productos>
```

# Elements-Nomenclatura

Els noms :

- Han de començar amb una lletra o un guió baix
- No poden començar amb les lletres XML. Es poden fer servir totes les paraules (excepte xml)
- Sí poden contenir lletres, dígits, guions, guions baixos i punts
- No poden tenir espais.

# Elements-Nomenclatura- Recomanacions

- Noms descriptius : <persona>, <Nom>, <cognom>.
- Noms curts i simples, <titulo\_libro> no com aquest: <mi\_titulo\_de\_un\_libro>.
- Evitar "-" (alguns programes poden pensar que és una resta)
- Evitar "." ( propietat a OOP)
- Evitar ":" (reservat pels espais de noms)
- Evitar les lletres no angleses(á,Á,ñ,Ñ..) encara que sí están permeses, però poden tenir incompatibilitats amb alguns programes que no les reconeixen.
- Es permet un espai en blanc o un salt de línea darrera del nom de l'etiqueta (davant no).

```
<ciudad >Pamplona</ciudad>
```

<Ciudad>Pamplona</ciudad>

<día>18</dia>

<mes>6<mes/>

<ciudad>Pamplona</finciudad>

<\_rojo>

<2colores>Rojo y Naranja</2colores>

< Aficiones >Cine, Bailar, Nadar</ Aficiones >

<persona><nombre>Elsa</persona></nombre>

<color favorito>azul</color favorito>

**Incorrecte**

<Ciudad>Pamplona</Ciudad>

<día>18</día>

<mes>6</mes>

<ciudad>Pamplona</ciudad>

<\_rojo/>

<colores2>Rojo y Naranja</colores2>

<Aficiones >Cine, Bailar, Nadar</Aficiones >

<persona><nombre>Elsa</nombre></persona>

<color\_favorito>azul</color\_favorito>

**Recomanació**

# Estils que es fan servir

Style	Example
Lower case	<firstname>
Upper case	<FIRSTNAME>
Underscore	<first_name>
Pascal case	<FirstName>
Camel case	<firstName>

# XML extensible

Partim d'una aplicació que extrau les dades d'un document XML com el següent:

```
<note>  
  <to>Tove</to>  
  <from>Jani</from>  
  <body>Don't forget me this weekend!</body>  
</note>
```



```
<note>  
  <date>2008-01-10</date>  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>
```

Si es fan canvis en el XML l'aplicació pot continuar funcionant igual, ja que continuà trobant els elements pels quals estava definit.

# Atributs

<year>2005</year>

Pot tenir text i atributs:

```
<book category="children">  
  <title lang="en">Harry Potter</title>  
  <author>J K. Rowling</author>  
</book>
```

L'atribut serveix per donar informació extra sobre l'element que el conté.

Per exemple, per desar les dades: codi:47P, nom: falda, color:vermell,preu:14.50

Representació XML:

```
<producte codi="47P">  
  <nom color="vermell" preu="14.50">falda</nom>  
</producte>
```



# Atributs Sintaxis

`<person gender="female">` ok    `<person gender='female'>` ok

`<gangster name='George "Shotgun" Ziegler'>`

`<gangster name="George &quot;Shotgun&quot; Ziegler">` ok

- Tots els atributs d'un element han de ser únics.

~~`<datos x="3" x="4" y="5"/>`~~

`<datos x="3" X="4" y="5"/>`

# Atributs Elements o atributs?

```
<person gender="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<person>
  <gender>female</gender>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Informació

```
<producte codi="47P">
  <nom color="vermell" preu="14.50">falda</nom>
</producte>
```

```
<producte>
  <codi> 47P</codi>
  <nom color="vermell" preu="14.50">falda</nom>
</producte>
```

# Atributs

```
<note date="2008-01-10">  
  <to>Tove</to>  
  <from>Jani</from>  
</note>
```

```
<note>  
  <date>2008-01-10</date>  
  <to>Tove</to>  
  <from>Jani</from>  
</note>
```

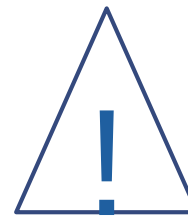
```
<note>  
  <date>  
    <year>2008</year>  
    <month>01</month>  
    <day>10</day>  
  </date>  
  <to>Tove</to>  
  <from>Jani</from>  
</note>
```

Els atributs no poden contenir múltiples valors, els elements sí.

Els atributs no poden tenir estructura d'arbre, els elements sí.

Els atributs són difícils d'expandir després si es volen fer canvis.

```
<note day="10" month="01" year="2008"  
to="Tove" from="Jani" heading="Reminder"  
body="Don't forget me this weekend!">  
</note>
```



# Atributs

- Els atributs (per ex id) poden servir per identificar elements XML com els id de l'HTML.

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

Ens serveix per identificar les diferents notes, és a dir és una dada de les dades (metadades) , no són dades per si mateixes.

**Recomanació: Les metadades s'han de desar com atributs i les dades en si com a elements.**

# Espai de noms

- Si volem fer servir diferents XML pot passar:

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

Conflicte de noms



```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

# Espai de noms

- És una recomanació W3C per proporcionar elements i atributs amb nom únic en un arxiu XML.
- Un arxiu XML pot contenir noms d'elements o atributs amb origen de més d'un vocabulari XML.
- Si a cada vocabulari se li dóna un espai de noms, un espai propi es resó el conflicte de noms que pot provocar els elements o atributs amb el mateix nom.
- És defineix fent referència a una URI.
- No es tracta de fer servir la URI com un enllaç, ni ha de tenir contingut, només es fa servir perquè el nom sigui únic.

# Espai de noms - sintaxi

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

Per exemple: aquí es defineixen dos espais de noms amb els prefixes e1 i e2.

```
<e1:ejemplo xmlns:e1="http://www.abrirllave.com/ejemplo1"  
  xmlns:e2="http://www.abrirllave.com/ejemplo2">
```

Les URI no tenen perquè tenir res, només serveixen per ser únics.

# Espai de noms - Exemples

- Si volem posar en un XML aquests dos

```
<carta>
  <palo>Corazones</palo>
  <numero>7</numero>
</carta>
```

```
<carta>
  <carnes>
    <filete_de_tenera precio="12.95"/>
    <solomillo_a_la_pimienta precio="13.60"/>
  </carnes>
  <pescados>
    <lenguado_al_horno precio="16.20"/>
    <merluza_en_salsa_verde precio="15.85"/>
  </pescados>
</carta>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<e1:ejemplo xmlns:e1="http://www.abrirllave.com/ejemplo1"
  xmlns:e2="http://www.abrirllave.com/ejemplo2">

  <e1:carta>
    <e1:palo>Corazones</e1:palo>
    <e1:numero>7</e1:numero>
  </e1:carta>

  <e2:carta>
    <e2:carnes>
      <e2:filete_de_tenera precio="12.95"/>
      <e2:solomillo_a_la_pimienta precio="13.60"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="16.20"/>
      <e2:merluza_en_salsa_verde precio="15.85"/>
    </e2:pescados>
  </e2:carta>

</e1:ejemplo>
```



# Espai de noms - Exemples

```
<?xml version="1.0"?>
<cli:cliente xmlns:cli='http://es.wikipedia.org/wiki/Espacio_de_nombres_XML/cliente'
             xmlns:ped='http://es.wikipedia.org/wiki/Espacio_de_nombres_XML/pedido'>
  <cli:numero_ID>1232654</cli:numero_ID>
  <cli:nombre>Fulanito de Tal</cli:nombre>
  <cli:telefono>99999999</cli:telefono>
  <ped:pedido>
    <ped:numero_ID>6523213</ped:numero_ID>
    <ped:articulo>Caja de herramientas</ped:articulo>
    <ped:precio>187,90</ped:precio>
  </ped:pedido>
</cli:cliente>
```

Els espais de noms es poden definir en cada element o des de l'arrel com abans.

```
<?xml version="1.0" encoding="UTF-8"?>
<e1:ejemplo xmlns:e1="http://www.abrirllave.com/ejemplo1">

  <e1:carta>
    <e1:palo>Corazones</e1:palo>
    <e1:numero>7</e1:numero>
  </e1:carta>

  <e2:carta xmlns:e2="http://www.abrirllave.com/ejemplo2">
    <e2:carnes>
      <e2:filete_de_ternera precio="12.95"/>
      <e2:solomillo_a_la_pimienta precio="13.60"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="16.20"/>
      <e2:merluza_en_salsa_verde precio="15.85"/>
    </e2:pescados>
  </e2:carta>

</e1:ejemplo>
```

- Definir diferents espais de noms pels elements.
- I cap per algun element (="" ).

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo xmlns="http://www.abrirllave.com/ejemplo1">

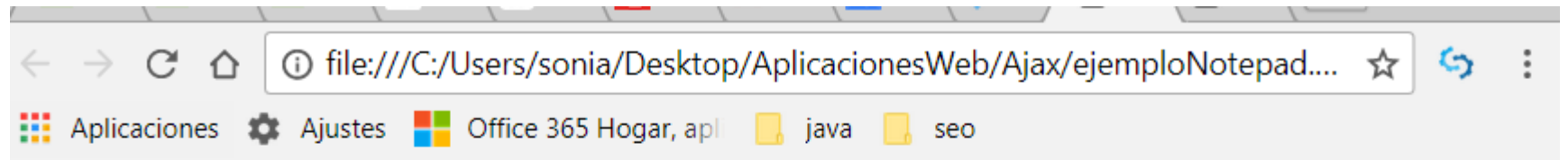
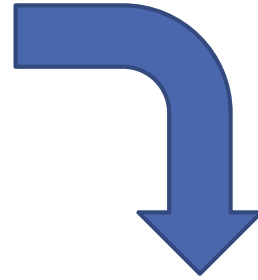
    <carta>
        <palo>Corazones</palo>
        <numero>7</numero>
    </carta>

    <carta xmlns="http://www.abrirllave.com/ejemplo2">
        <carnes>
            <filete_de_tenera precio="12.95"/>
            <solomillo_a_la_pimienta precio="13.60"/>
        </carnes>
        <pescados xmlns="">
            <lenguado_al_horno precio="16.20"/>
            <merluza_en_salsa_verde precio="15.85"/>
        </pescados>
    </carta>

</ejemplo>
```

# Visualització al Navegador

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <element nombre="primero" />
  <element nombre="segundo" />
</root>
```



This XML file does not appear to have any style information associated with it. The document tree is shown below.

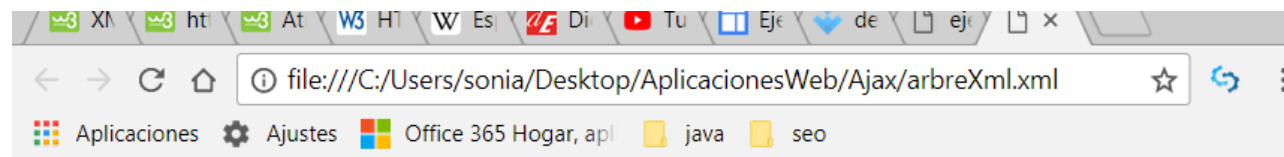
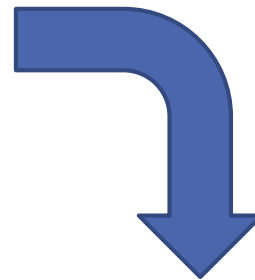
```
▼ <root>
  <element nombre="primero"/>
  <element nombre="segundo"/>
</root>
```

# Visualització al Navegador(errors)

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>  <!-- Element arrel-->
  <book category="cooking">
    <!-- Element book que té quatre
    elements secundaris
    title,author,yera,price-->
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>

    <year> any <2005</year>
    <year> any &lt;2005</year>

    <Price>29.99</price> incorrecte
    <Price>29,99</Price> correcte
  </book>
</bookstore>
```



## This page contains the following errors:

error on line 14 at column 17: StartTag: invalid element name

## Below is a rendering of the page up to the first error.

Everyday Italian Giada De Laurentiis 2005 30.00 Harry Potter J K. Rowling any

# Exercicis XML ESTRUCTURA

# Associar un arxiu CSS a un document XML

```
<?xml-stylesheet type="text/css" href="miestilo.css"?>
```

# XML amb CSS (catalog.css i catalog.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="catalog.css"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
</CATALOG>
```

```
CATALOG {
  background-color: #ffffff;
  width: 100%;
}
CD {
  display: block;
  margin-bottom: 30pt;
  margin-left: 0;
}
TITLE {
  display: block;
  color: #ff0000;
  font-size: 20pt;
}
ARTIST {
  display: block;
  color: #0000ff;
  font-size: 20pt;
}
COUNTRY, PRICE, YEAR, COMPANY {
  display: block;
  color: #000000;
  margin-left: 20pt;
}
```

Empire Burlesque  
Bob Dylan

USA  
Columbia  
10.90  
1985

Hide your heart  
Bonnie Tyler

UK  
CBS Records  
9.90  
1988

No recomanable XML amb CSS.  
Millor JS o XSLT .

Sonia Sánchez Sierra



# Exercici

- En un XML es desa la informació del nom d'un animal i el número de pots que té.
- Volem una sortida al navegador del tipus:

|perro<sub>4</sub> pato<sub>2</sub> ballena<sub>0</sub>

# Solució

(animales.css, animales.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="animales.css"?>
<animales>
  <animal>
    <nombre>perro</nombre>
    <patas>4</patas>
  </animal>
  <animal>
    <nombre>pato</nombre>
    <patas>2</patas>
  </animal>
  <animal>
    <nombre>ballena</nombre>
    <patas>0</patas>
  </animal>
</animales>
```

```
nombre{color:blue;font-size:40px}
patas{color:red;font-size:22px}
```

# Exercicis XML CSS Errores

# Parser (analitzador)

- Les etiquetes del XML fan possible poder tenir una representació interna com el **DOM d'HTML**.
- **Parser**:= script,mòdul,biblioteca o programa que s'encarrega de transformar **un arxiu de text en una representació interna**.
- **EL XML DOM** (Document Object Model) defineix les **propietats i mètodes** per accedir i editar XML.
- Per poder accedir al XML s'ha de carregar el **DOM XML**.
- Els **navegadors** tenen un **parser integrat** que pot convertir **el text en un objecte DOM XML**.

```
<!DOCTYPE html>
<html>
<body>

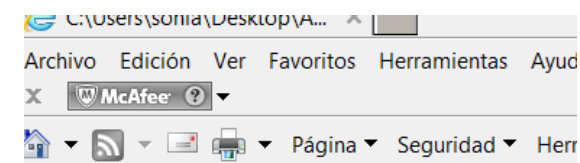
<p id="demo"></p>

<script>
var parser, xmlDoc;
var text = "<bookstore><book>" +
"<title>Everyday Italian</title>" +
"<author>Giada De Laurentiis</author>" +
"<year>2005</year>" +
"</book></bookstore>";

parser = new DOMParser();
xmlDoc = parser.parseFromString(text, "text/xml");

document.getElementById("demo").innerHTML =
xmlDoc.getElementsByTagName("year")[0].childNodes[0].nodeValue;
</script>

</body>
</html>
```



2005

Ejemplo de parser 'interno'

analizador.html

- Creem un DOMParser , un analitzador XML DOM : parser= `new DomParser()`
- Parsear XML des d'una cadena de text ,mètode: `parseFromString()` per obtenir un objecteDOM XML

```
var parser = new DOMParser();  
var xmlDoc = parser.parseFromString(text,"text/xml");
```

```
var text = "<bookstore><book>" +  
"<title>Everyday Italian</title>" +  
"<author>Giada De Laurentiis</author>" +  
"<year>2005</year>" +  
"</book></bookstore>";
```

- Hi ha tres resultats diferents possibles, depenent del tipus MIME donat:
  - ✓ application/xml, objecte resultant serà un Document
  - ✓ image/svg+xml objecte resultant serà un SVGDocument
  - ✓ text/html objecte resultant serà un HTMLDocument.

```
var parser = new DOMParser();  
var doc = parser.parseFromString(stringContainingXMLSource, "application/xml");  
// returns a Document, but not a SVGDocument nor a HTMLDocument  
  
parser = new DOMParser();  
doc = parser.parseFromString(stringContainingXMLSource, "image/svg+xml");  
// returns a SVGDocument, which also is a Document.  
  
parser = new DOMParser();  
doc = parser.parseFromString(stringContainingHTMLSource, "text/html");  
// returns a HTMLDocument, which also is a Document.
```

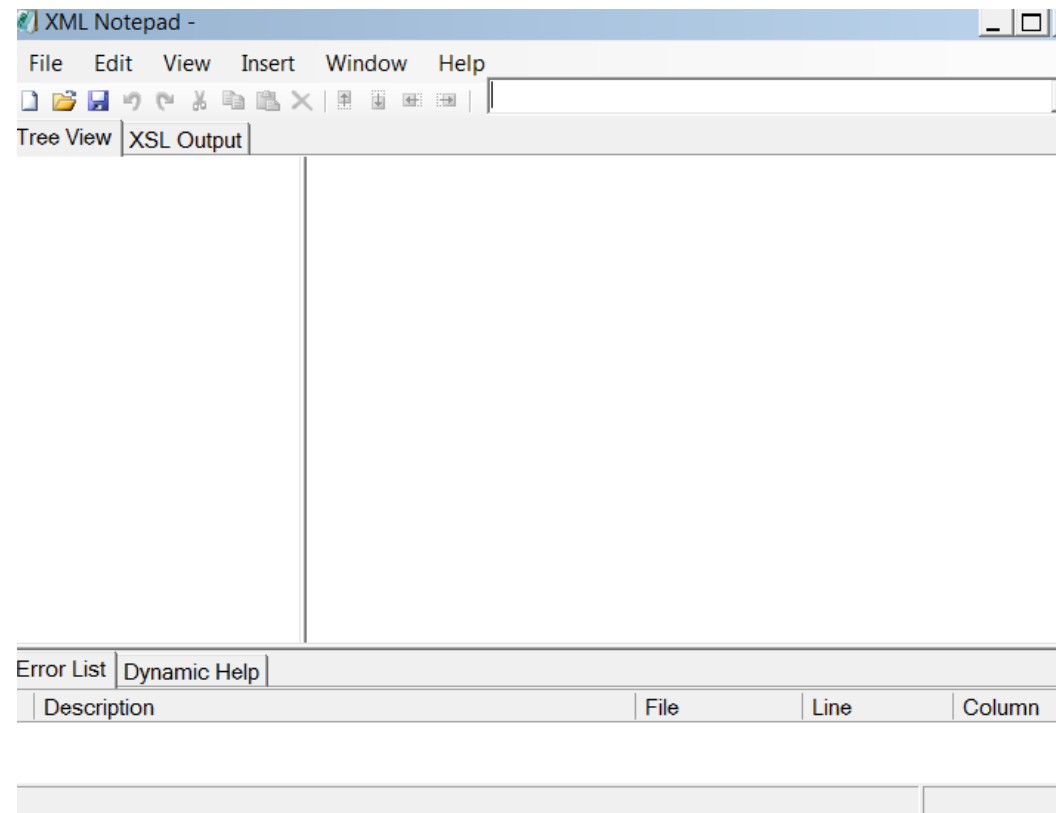
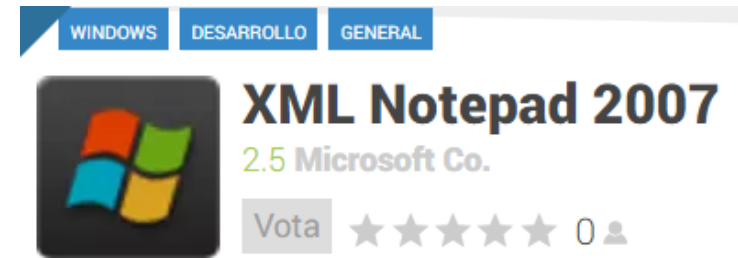
# DOM

- HTML DOM defineix una forma estàndard per accedir i manipular documents HTML. Presenta un document HTML como una estructura d'arbre.
- XML DOM defineix una forma estàndard per accedir i manipular documents XML. Presenta un document XML como una estructura d'arbre.

```
/*accès HTML DOM*/  
document.getElementById("demo").innerHTML =  
xmlDoc.getElementsByTagName("year")[0].childNodes[0].nodeValue;  
/*accès XML DOM*/
```

# XML Notepad

- <https://xml-notepad-2007.uptodown.com/windows>





Anem a fer el següent arxiu xml:

Definim la versió:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <element nombre="primero" />
  <element nombre="segundo" />
</root>
```

