

# CREACIÓ DE DOCUMENTS XML

UF 1: PROGRAMACIÓ AMB XML - PART 2

# XML

“eXtensible Markup Language”

Estàndard del W3C per documents.

# XML

- » Defineix una sintaxi genèrica per marcar les dades textuais amb valors intel·ligibles per humans.
- » És una forma de donar format als documents prou flexible per ser personalitzada per diferents dominis: pàgines web, impressores, bases de dades, etc...
- » Està pensat perquè tothom el pugui fer servir en totes les possibles àrees d'interès i per sempre més.

# XML

» XML NO és:

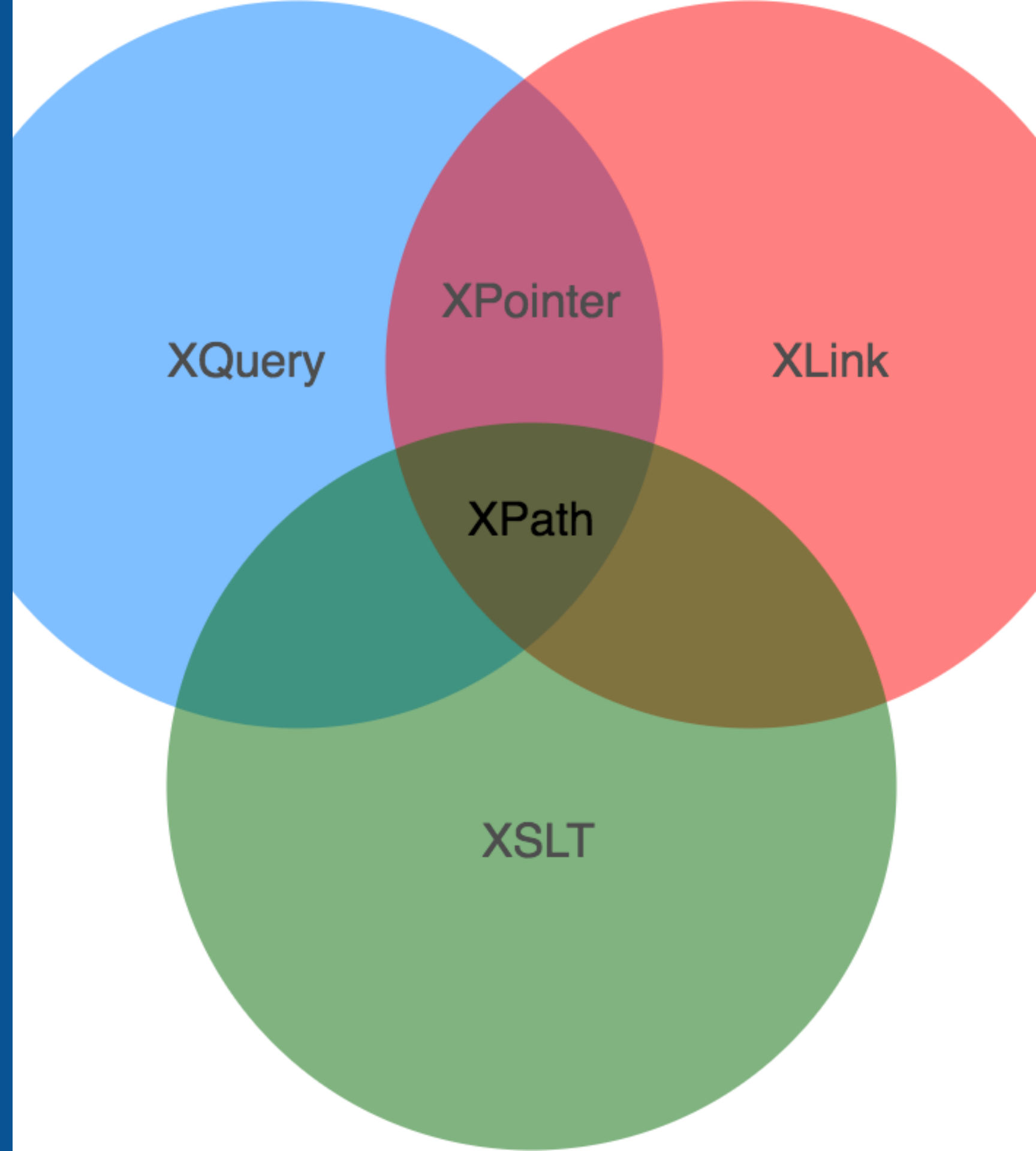
» No és un llenguatge de programació

» No és un protocol de xarxa

» No és una base de dades

# XML

XML no és només un llenguatge de llenguatges, també és un conjunt de tecnologies que permet la seva representació i processament

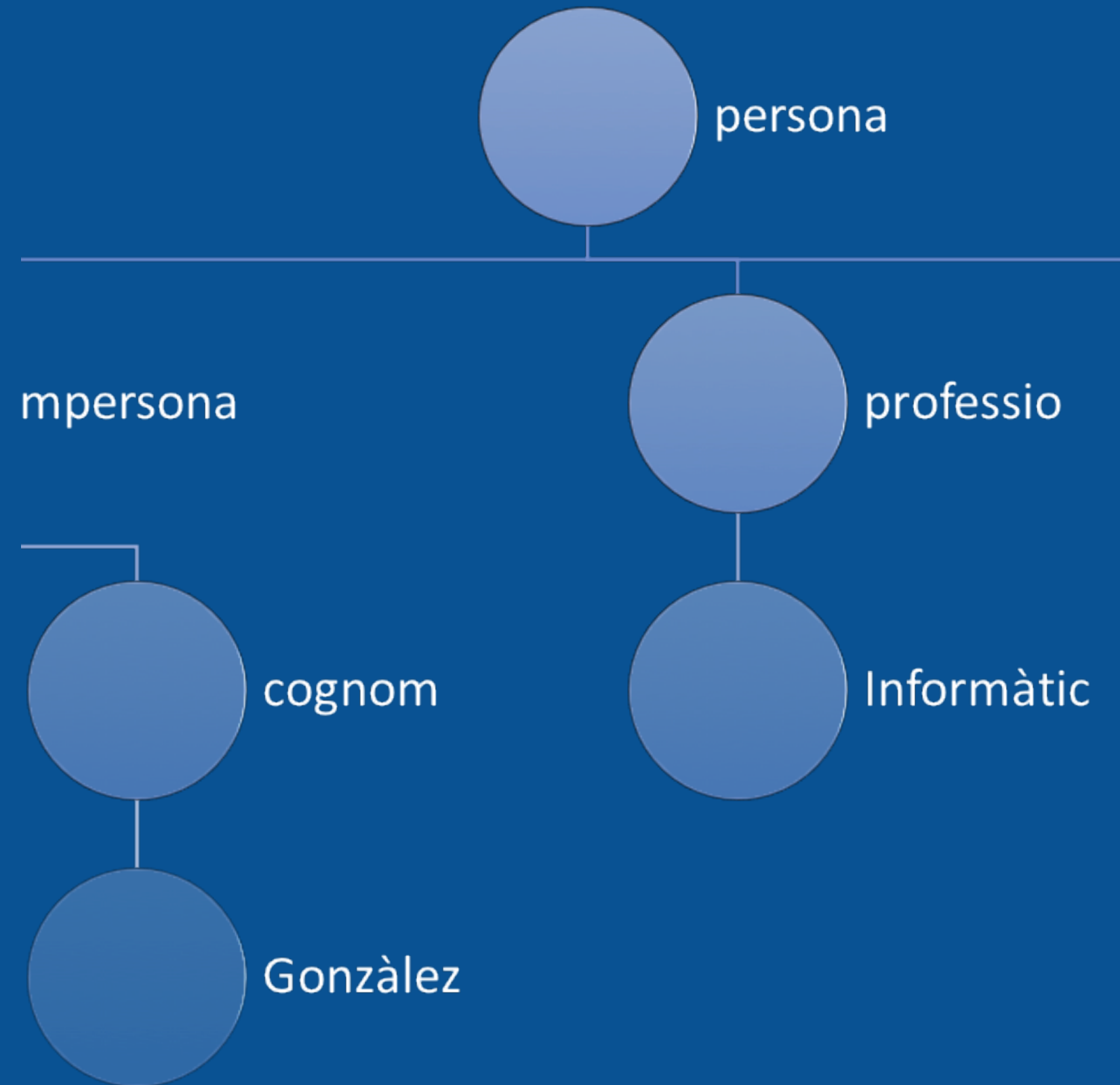


# XML

- » Les dades d'un document XML es poden representar en arbres
- » Tots els documents XML tenen una etiqueta pare que no és fill de ningú
- » Les fulles de l'arbre són les dades contingudes en el document

# XML: EXEMPLE

```
<persona>  
  <nompersona>  
    <nom>Carles</nom>  
    <cognom>González</cognom>  
  </nompersona>  
  <professio>  
    Informàtic  
  </professio>  
  <professio>  
    Professor  
  </professio>  
</persona>
```



# ANÀLISI XML

- » L'objectiu principal de tenir regles a l'hora d'escriure documents XML és que puguin ser llegits i interpretats per un programa d'ordinador
- » L'anàlisi de fitxers XML es fa amb analitzadors (parsers)
  - » Comproven que es compleixen les regles d'XML
  - » Comproven que el document està “ben format”
  - » També poden comprovar que el document és “vàlid”



# ANÀLISI XML

- » Generalment consisteix en dues tasques:
  - » Comprovar-ne la correctesa
  - » Comprovar-ne la validesa

# COMPROVAR-NE LA CORRECTESA

- » Consisteix en comprovar que el document segueix les regles de creació de documents XML
- » S'assegura que el document està “Ben format”

# COMPROVAR-NE LA VALIDESA

- » Consisteix en comprovar que el document compleix amb les especificacions del nostre vocabulari
- » S'assegura que el document és “vàlid”

# ERRORS

- » XML deixa ben clar com s'han de tractar els errors que es produeixin en els documents:
- » La gestió es diferent en errors lleus i greus

# ERRORS LLEUS

“Violació d'algunes de les regles que estan marcades com a recomanació d'XML”

El programa pot intentar continuar

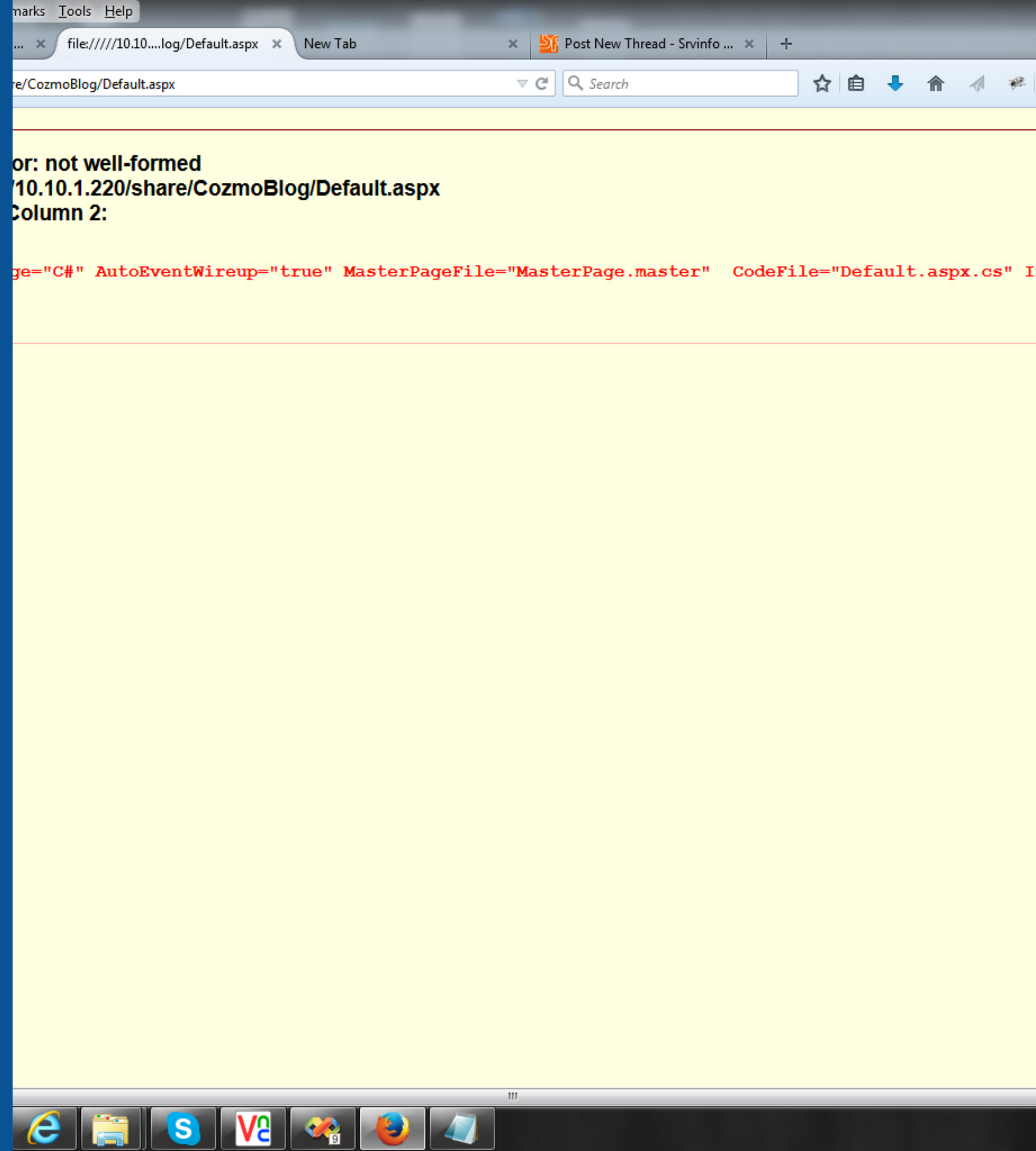
# ERRORS GREUS

“No es compleix alguna de les regles obligatòries: no seguir la sintaxi, etc...”

El programa ha d'aturar-se immediatament i no continuar amb el procés

# ANÀLISI XML

Podem comprovar el compliment de les regles amb un navegador web



# ANALITZADORS XML

» Hi ha molts analitzadors de documents i molts d'ells són lliures:

» Xmllint:

» Forma part de la llibreria libXML

» Apache Xerces:

» Està inclòs dins del projecte XML de la fundació Apache. Suport per Java, C i Perl



# ANALITZADORS XML

» Expat:

» Expat és una llibreria d'anàlisi XML feta en C

» Microsoft XML Parser (MSXML)

» Eina de Microsoft gratuïta que està inclosa en el seu navegador Internet Explorer

# XMLLINT

- » Analitzador i validador de documents XML
- » Si tot va bé ens mostra el document
- » Si hi ha errors ens els donarà informació per corregir-ho

```
$ xmllint persona.xml
```

```
<?xml version="1.0"?>
```

```
<persona>
```

```
<nompersona>
```

```
<nom>Xavier</nom>
```

```
<cognom>Sala</cognom>
```

```
</nompersona>
```

```
</persona>
```

# EXPAT

» Amb el paquet d'Ubuntu hi tenim el programa `xmlwf` que ens permet analitzar documents xml

» Aquest programa fa servir la llibreria de C Expat:

```
$ xmlwf fitxer.xml
```

```
error.xml:6:2: mismatched tag
```

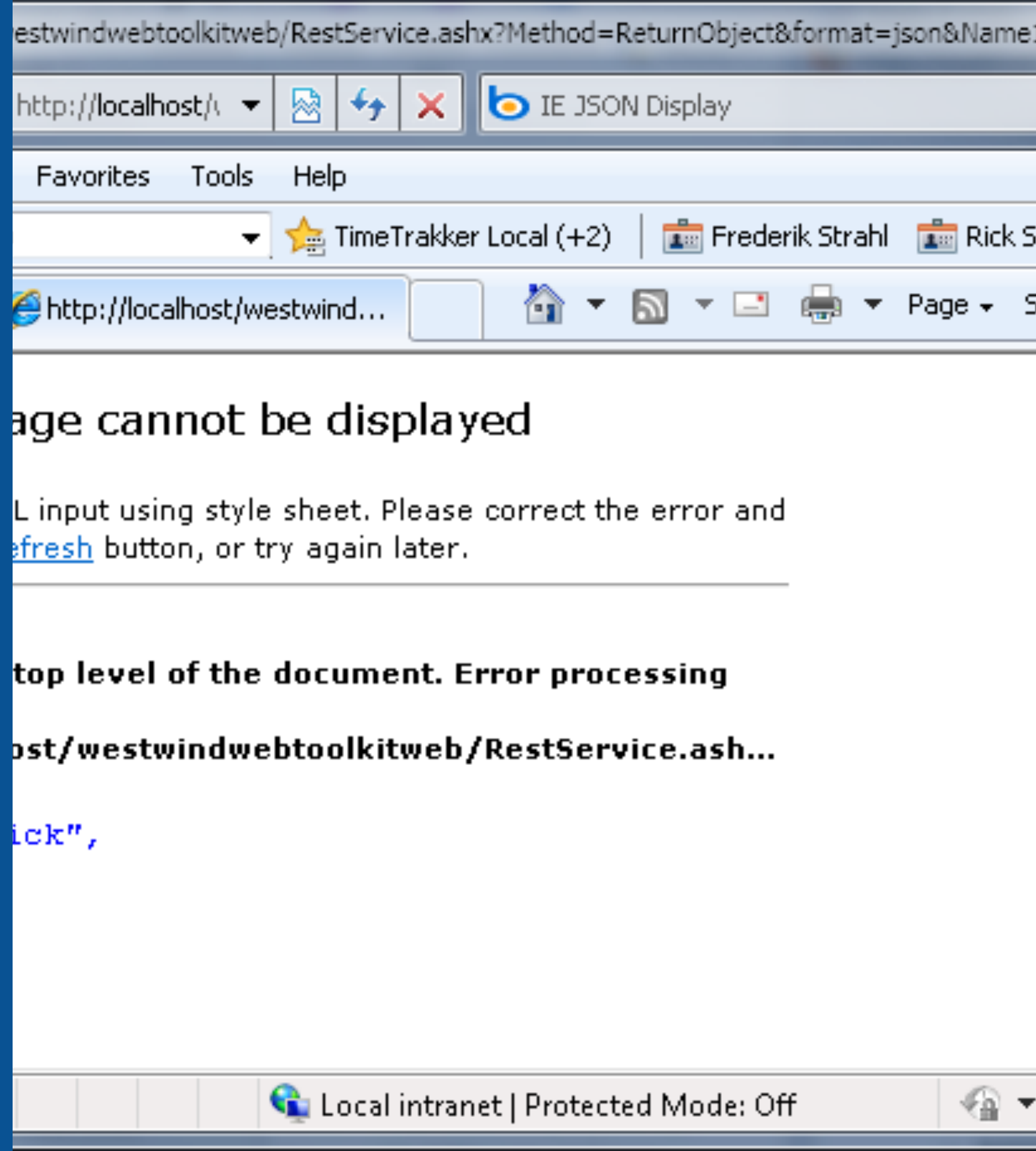
```
$ xmlwf fitxer.xml
```

```
$
```

» Només analitza la correctesa però no fa validacions

# MSXML

- » El Microsoft XML Parser dóna suport per:
  - » Accés HTTP amb servidor segur
  - » Implementa XSL i Xpath. S'ajusta als estàndards



## EDITAR XML

- » XML es pot crear amb qualsevol editor de textos
- » Alguns fins i tot ofereixen ressaltat de colors
- » Visual Studio Code, Atom, Gedit, Kate

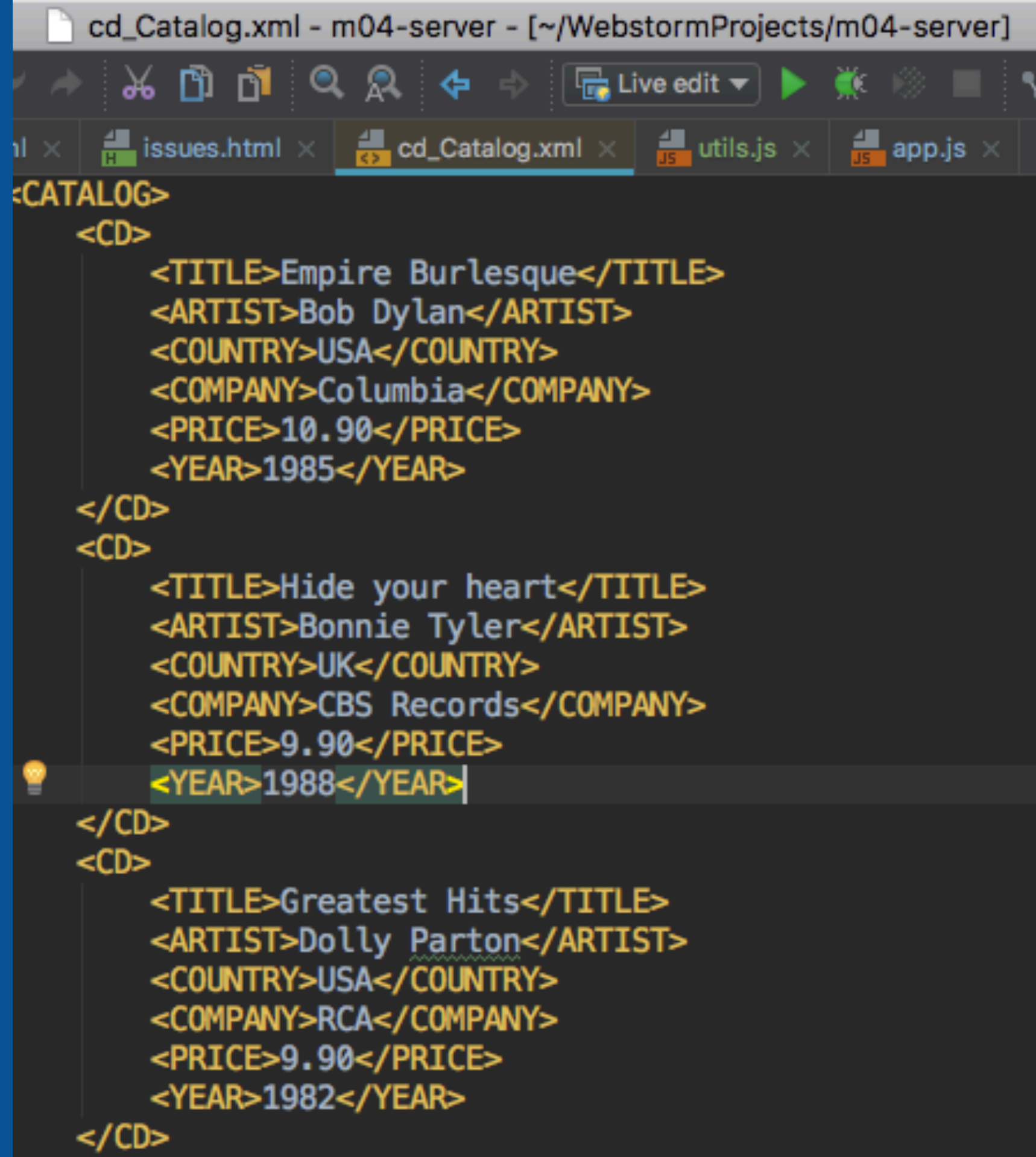
```
cd_Catalog.xml — reveal.js
creacio.html  M+ uf2-creacio.md  cd_Catalog.xml
CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
```

# EDITORS XML

- » Tot i així sempre ajudarà fer servir algun editor amb suport XML
- » Aquests editors sovint ofereixen algunes característiques extremes
  - » Auto emplenat
  - » Validació de documents
  - » Creació d'esquemes
  - » Transformacions, etc...

# EDITORS XML

- » Hi ha editors com XML Copy Editor específics per treballar en XML
- » Sovint els IDEs de programació tenen suport:
  - » IntelliJ
  - » Microsoft Visual Studio
  - » Webstorm



The screenshot shows a code editor window titled "cd\_Catalog.xml - m04-server - [~/WebstormProjects/m04-server]". The editor displays XML code for a CD catalog. The code is as follows:

```
<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>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>RCA</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
</CATALOG>
```

# EDITORS XML

- » Si tenim que fer edicions complexes o utilitzar eines com XPath podem fer servir editors especialitzats com:
  - » Altova XMLSpy
  - » oXigen XML
  - » Liquid XML Studio
  - » Stylus Studio 6 XML
  - » XMLPad



# DOCUMENTS XML REGLES

```
name="Facebook Members WordPress Plugin" image="image1.png" price="39.50" description="Large"
w_detail image="image1.png">Image1</aiow_detail>
w_detail details="This is detailed description on WordPress Plugin">Mini Description</aiow_detail>
description="Large">
w_detail image="image2.jpg">Image2</aiow_detail>
w_detail details="This is detailed description on WordPress Plugin">Detailed Description</aiow_detail>
ny_item>
ny_item name="Facebook Members WordPress Plugin">
number>FM01</item_number>
price>42.50</price>
description="Small">
w_detail image="image3.jpg">Red</aiow_detail>
w_detail details="this is WordPress Plugin">
no_of_downloads name="facebook-members">More than 400k</no_of_downloads>
w_detail>
ny_item>
name>
```

This is what I want in Java



More than 400k

# INTRODUCCIÓ

- » Els documents XML contenen text, etiquetes i atributs

```
<nom carrec="professor"> Carles Gonzàlez </nom>
```

- » Etiquetes: <nom> </nom>
- » Atributs: carrec="professor"
- » Text: Carles Gonzàlez

No es pressuposa res de com es representaran les dades

# REGLES BÀSIQUES

- » Tots els documents tenen una etiqueta arrel
- » Totes les etiquetes s'han de tancar
- » Les etiquetes han d'estar niades correctament
- » Els noms de les etiquetes han de ser correctes
- » Les majúscules i minúscules són diferents
- » Els valors dels atributs han d'estar entre cometes

# ETIQUETA ARREL

» Tots els documents han de tenir una etiqueta arrel que contingui totes les altres

» En el següent exemple seria `<persona>`

```
<persona>
```

```
  <nom>Pepe Pérez</nom>
```

```
  <nom>Fulano de Tal</nom>
```

```
</persona>
```

» Aquest no seria un document XML correcte

```
<nom>Pepe Pérez</nom>
```

```
<nom>Fulano de Tal</nom>
```

» L'únic que pot estar fora de l'arrel són els comentaris i instruccions de procés

# LES ETIQUETES S'HAN DE TANCAR

» Cada una de les etiquetes dins d'un document XML s'ha de tancar.

```
<article>Disquet</article>
```

» No es poden fer coses com les que permet HTML

```
<article nom="disquet">
```

» Si tenim una etiqueta sense dades el podem representar amb el tancament “/>”:

```
<article nom="disquet" />
```

# CORRECTAMENT NIADES

» Si es comencen dos etiquetes aquestes s'han de niar correctament

```
<inventari>  
  <article>  
    Disquet  
  </article>  
</inventari>
```

» No es poden tancar de qualsevol forma

```
<inventari>  
  <article>  
    Disquet  
  </inventari>  
</article>
```

Està tancat en ordre invers i per tant és incorrecte

# NOMS CORRECTES

- » Els noms de les etiquetes han de ser correctes:
  - » Han de començar per una lletra de l'alfabet o un subratllat, guió o dos punts
  - » No poden tenir espais
  - » No poden començar per la paraula 'xml' sigui com sigui escrita
  - » No hi pot haver cap espai després de l'inici de la etiqueta '<'

# NOMS CORRECTES

## CORRECTES

<Ciutat> </Ciutat>

<Item12> </Item12>

<casa-Nova> </casa-Nova>

<element > </element >

<\_Carai> </\_Carai>

## INCORRECTES

<XMLElement> </XMLElement>

<10sset> </10sset>

< Item> </ Item>

<Pepe Pérez> </Pepe Pérez>

<\$Diners> </\$Diners>



# MAJÚSCULES != MINÚSCULES

Per XML les majúscules són lletres diferents que les minúscules

Per tant aquestes dues són diferents:

```
<Article> Disquet </Article>
```

```
<Article> Disquet </article>
```

Això és incorrecte:

```
<Article> Disquet </article>
```

# VALORS ENTRE COMETES

» Els valors dels atributs han d'estar entre cometes fins i tot si són números

```
<Article quantitat="3">Disquet</Article>
```

» No es pot fer això:

```
<Article quantitat=3>Disquet</Article>
```

» Els atributs sempre han de tenir algun valor, no es poden tenir atributs sense valor

No importa si són cometes simples o dobles

# DOCUMENTS XML CREATE DOCUMENTS

```
name="Facebook Members WordPress Plugin" data-bbox="0 65 1000 100"/>
y_it data-bbox="0 100 1000 135"/>
_number data-bbox="0 135 1000 170"/>
>39.99</price> data-bbox="0 170 1000 205"/>
description="Large" data-bbox="0 205 1000 240"/>
w_detail image="image2.jpg">Image2</aiow_detail> data-bbox="0 240 1000 275"/>
w_detail details="This is detailed description on WordPress Plugin">Detailed Description</aiow_ data-bbox="0 275 1000 310"/>
e> data-bbox="0 310 1000 345"/>
description="Large"> data-bbox="0 345 1000 380"/>
w_detail image="image2.jpg">Image2</aiow_detail> data-bbox="0 380 1000 415"/>
w_detail details="This is detailed description on WordPress Plugin">Detailed Description</aiow_ data-bbox="0 415 1000 450"/>
e> data-bbox="0 450 1000 485"/>
ny_item> data-bbox="0 485 1000 520"/>
y_item name="Facebook Members WordPress Plugin"> data-bbox="0 520 1000 555"/>
_number>FM01</item_number> data-bbox="0 555 1000 590"/>
e>42.50</price> data-bbox="0 590 1000 625"/>
description="Small"> data-bbox="0 625 1000 660"/>
w_detail image="image3.jpg">Red</aiow_detail> data-bbox="0 660 1000 695"/>
w_detail details="this is WordPress Plugin"> data-bbox="0 695 1000 730"/>
o of downloads name="facebook-members">More than 400k</no_of_downloads> data-bbox="0 730 1000 765"/>
ow_detail> data-bbox="0 765 1000 800"/>
e> data-bbox="0 800 1000 835"/>
ny_item> data-bbox="0 835 1000 870"/>
_name> data-bbox="0 870 1000 905"/>

```

This is what I want in Java

More than 400k

# LA DECLARACIÓ XML

- » Els documents XML comencen amb la declaració que indica quina versió estem fent servir d'XML
- » La declaració no és totalment obligatoria però és recomanable

```
<?xml version="1.0" ?>
```

- » Atribut "version": Generalment es fa servir la versió "1.0" a menys que hi hagi causes de força major
- » La versió "1.1" aporta poques millores

# LA DECLARACIÓ XML

- » Atribut Encoding: Defineix el codi de caràcters que fem servir en el document
- » Com que XML està definit per un organisme internacional fa servir Unicode per defecte (UTF-8 i UTF-16) però en pot fer servir d'altres

FORMAT	LLENGUATGE	FORMAT	LLENGUATGE
US-ASCII	Anglès	ISO-8859-3	Sud d'Europa
Windows-1250	Europa central	UTF-16	UCS Comprimat
ISO-8859-2	Est d'Europa	ISO-8859-4	Nord d'Europa
UTF-8	Unicode comprimit	ISO-8859-1	Oest d'Europa
Windows-1251	Ciril·lic	ISO-2022-JP	Japonès

# LA DECLARACIÓ XML

- » Atribut standalone: Defineix si el document XML no necessita un altre fitxer o si que el necessita
- » `standalone="yes"`: Significa que el document és complet i per tant no li calen altres documents
- » `standalone="no"`: Vol dir que el document no es pot entendre per si sol.
- » Generalment degut a que li falta algun fitxer per validar-lo

# DECLARACIÓ XML

» Per tant la declaració d'un fitxer podria ser con la següent:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

» Aquesta declaració ens dona la següent informació:

» Segueix XML versió 1.0

» Utilitza Caràcters Unicode

» Necessita un DTD per ser validat

# ETIQUETES

- » Les etiquetes van entre els símbols “<” i “>”
- » Les etiquetes són creades per l'autor i han de ser autoexplicatives
- » Descriuen les dades que contenen
- » Poden tenir tants atributs com faci falta

» Exemple:

```
<ciutat>Barcelona</ciutat>
```

» Queda clar que el text és una ciutat



# ATRIBUTS

» Les etiquetes poden tenir atributs que en donen informació

```
<nom carrec="professor">
```

» Els atributs són un parell format per `clau=valor`

» El nom dels atributs ha de ser un nom XML vàlid i no repetit

» N'hi poden haver tants com calgui

» Sempre han de tenir alguna cosa assignada

» Valid: `<nom carrec="">`

» NO Valid: `<nom carrec>`

# ATRIBUTS ESPECIALS

» Tenim dos atributs predefinits:

» `xml:lang`: Permet especificar en quin idioma està el contingut. El valor ha de ser un codi d'idioma ISO

```
<missatge xml:lang="ca">Hola!</missatge>
```

```
<missatge xml:lang="fr">Salut!</missatge>
```

» `xml:space`: Per forçar a que es conservin els espais en blanc: 'default' o 'preserve'

```
<missatge xml:space="preserve">
```

```
    Hola
```

```
    Manolito!
```

```
</missatge>
```

# TEXT: ENTITATS

- » El text pot ser com es vulgui excepte uns quants caràcters que s'han de canviar i que es coneixen com entitats
  - » No hi ha regles sobre quina ha de ser la llargada del text
  - » Es pot escriure en qualsevol idioma del món
  - » Es preserven els espais en blanc

SIMBOL	TRANSCRIPCIÓ
<	&lt;
>	&gt;
“	&quot;
'	&apos;
&	&amp;

# TEXT

- » Podem posar el text en una secció CDATA si no volem canviar els caràcters

```
<valor>  
  <! [CDATA[  
    Si 5 > x llavors sumar 1 a x  
  ] ]>  
</valor>
```

- » Generalment es fa servir per
  - » grans llargades de text que requeririen molts canvis
  - » Incloure HTML o Javascript (condicions, ...)

# COMENTARIS

- » Es poden afegir comentaris en els documents XML
- » Els comentaris seran ignorats pels programes que processin el fitxer XML
- » Els comentaris estan entre els caràcters "<!--" i "-->" i poden ser de múltiples línies

```
<!-- Això és un comentari -->
```

```
<!--
```

```
I això també
```

```
és un comentari
```

```
-->
```

# INSTRUCCIONS DE PROCÉS

- » Les instruccions de procés ens deixen que programes externs puguin realitzar tasques
- » Les instruccions de procés es tanquen entre "<?" i "?>" i el primer que s'hi posa és quin programa les processa

```
<? processador funcio(valor) ?>
```

- » El programa és obligatori i en les dades s'hi pot posar qualsevol cosa
- » Des del punt de vista de XML no té cap sentit

# ESPAIS DE NOMS

- » Com que cadascú pot crear les etiquetes que vulgui
  - » Al combinar documents podem trobar-nos que tenim etiquetes repetides
  - » Tot i que tinguin el mateix nom poden significar coses diferents!
- » Això provocaria un conflicte que no podríem resoldre
  - » La solució és indicar-li a l'ordinador alguna forma de diferenciar-los

# ESPAIS DE NOMS

» Els espais de noms permeten definir a quin vocabulari pertany un element d'un document HTML

» Amb els espais de noms podem definir noms únics

```
<alumnes:nom>Manolito Porras</alumnes:nom>
```

```
<professors:nom>Filomena Pi</professors:nom>
```

» Ara queda clar a quin espai pertany cada una de les etiquetes “nom”



# ESPAIS DE NOMS

» Els espais de noms s'han de carregar abans de l'inici de les dades del document XML

```
<?xml version="1.0"?>
<classe xmlns:A="http://server/alumnes"
        xmlns:P="http://server/professors">
  <A:nom>Manolito Porras</A:nom>
  <P:nom>Filomena Pi</P:nom>
</classe>
```

» En la definició tenim

```
<classe xmlns:a="http://server/alumnes">
```

» xmlns: Paraula clau

» Prefixe o alias de l'espai de noms: a

» Nom de l'espai: http://server/alumnes

# ESPAIS DE NOMS

- » A pesar de la semblança dels noms amb les adreces web això només són noms
- » Si hi anem amb el navegador no hi trobarem cap document
- » Es van fer servir per aconseguir tenir noms d'espais de noms únics i com que les adreces web ja ho són...
- » Podem definir quin és l'espai de noms per defecte sense posar-li prefix

```
<classe xmlns="http://server/alumnes">
```

# ESPAIS DE NOMS

- » No cal definir tots els espais de noms en l'arrel
- » També es poden definir directament en els elements
- » També és vàlid pels seus descendents

```
<classe xmlns="http://server/alumnes">  
  <nom>Mariano Pardines</nom>  
  <nom xmlns="http://server/professors">  
    Federicu Pi  
  </nom>  
</classe>
```

# ESPAIS DE NOMS

» Per tant és el mateix:

```
<?xml version="1.0"?>
<classe xmlns:A="http://server/alumnes"
        xmlns:P="http://server/professors">
  <A:nom>Manolito Porras</A:nom>
  <P:nom>Filomena Pi</P:nom>
</classe>
```

```
<?xml version="1.0"?>
<classe xmlns="http://server/alumnes"
        xmlns:P="http://server/professors">
  <nom>Manolito Porras</nom>
  <P:nom>Filomena Pi</P:nom>
</classe>
```

```
<?xml version="1.0"?>
<classe xmlns="http://server/alumnes">
  <nom>Manolito Porras</nom>
  <nom xmlns="http://server/professors">Filomena Pi</nom>
</classe>
```

# ESPAIS DE NOMS

» Amb els espais de noms podem integrar les etiquetes de diferents llenguatges dins d'un document XML: SMIL, SVG, XHTML, ...

NOM	URL ESPAI DE NOMS
XML	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
XSL	<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>
XSL-FO	<a href="http://www.w3.org/1999/XSL/Format">http://www.w3.org/1999/XSL/Format</a>
XHTML	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>
SVG	<a href="http://www.w3.org/TR/svg">http://www.w3.org/TR/svg</a>