

1.- Qué és XML?

XML, "Extensible Markup Language" en anglès, és un llenguatge de marques utilitzat per emmagatzemar i intercanviar dades estructurades de manera legible tant per humans com per màquines.

Principals Característiques d'XML:

- **És un Llenguatge de Marques:** ja que **utilitza etiquetes per definir i estructurar les dades**. Les etiquetes són elements que envolten les dades i descriuen què representen. Per exemple, **<persona>** pot ser una etiqueta que envolta les dades d'una persona.
- **És Extensible:** En XML "extensible" vol dir **que podem definir les nostres pròpies etiquetes i estructures de dades**. Això permet la flexibilitat en la definició de formats de dades específics per a diferents aplicacions.
- **Es Jerarquitzat:** XML **segueix una estructura jeràrquica**, on les **etiquetes poden contenir altres etiquetes**, creant **una estructura en forma d'arbre**. Aquesta jerarquia ajuda a representar relacions complexes entre les dades.
- **Llegible pels Humans:** Les dades en un document XML **estan en un format llegible per humans**. Això facilita la inspecció i l'edició manual del contingut XML.
- **Interoperabilitat:** XML és un **estàndard amplament acceptat**, i la seua estructura és **comprensible per moltes aplicacions i plataformes diferents**. Això fa que sigui fàcil intercanviar dades entre sistemes diversos.
- **Usos Variats:** XML **es fa servir en una àmplia gamma d'aplicacions**, com ara el processament de textos, les bases de dades, la representació de documents HTML, el protocol per a serveis web, configuració de software, ... i molt més.
- **Compatibilitat amb Unicode:** XML **pot emmagatzemar text en diverses codificacions**, incloent **Unicode**, el que permet representar una àmplia gamma de caràcters de diferents idiomes i simbologies.

En resum, XML és un llenguatge de marques **versàtil i estandarditzat** que s'utilitza per **representar i intercanviar dades** estructurades de manera clara i flexible en una àmplia varietat d'aplicacions informàtiques.

• XML NO és:

- **No és un llenguatge de programació**
- **No és un protocol de xarxa**
- **No és una base de dades**

- XML **no és simplement un sistema de marques per estructurar dades**, sinó que també inclou tot un conjunt d'eines i **tecnologies relacionades que faciliten la creació, l'emmagatzematge, la transmissió i el processament de dades** en format XML.



→ **XPath i XQuery**: Són llenguatges utilitzats per seleccionar i consultar dades dins de documents XML. Permeten buscar, filtrar i extreure informació específica.

→ **XPointer (XML Pointer Language)**: llenguatge utilitzat per identificar parts específiques d'un document XML

→ **XLink (XML Linking Language)**: llenguatge que proporciona una forma estructurada d'afegir enllaços hipertextuals als documents XML

→ **DTD (Document Type Definition) i XML Schema**: Són mecanismes per definir la estructura i la validesa de les dades XML. Aquestes definicions permeten assegurar que les dades XML segueixin un format específic.

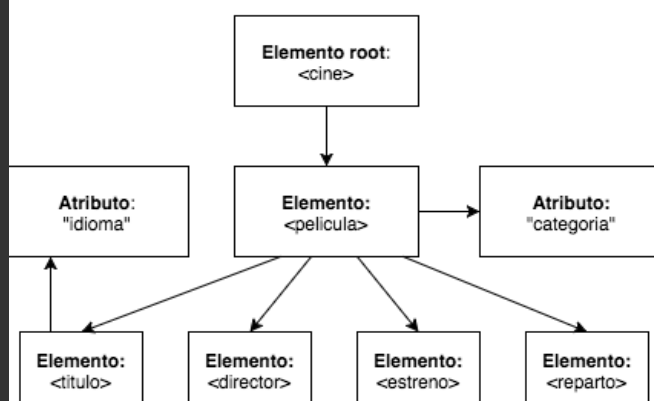
→ **XSL (Extensible Stylesheet Language)**: Per transformar documents XML en altres formats, com ara HTML, o per aplicar estils a documents XML per a la seva presentació.

I més...

En XML:

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

```
<?xml version="1.0" encoding="UTF-8"?>
<cine>
  <pelicula categoria="accion">
    <titulo idioma="ingles">Mad Max</titulo>
    <director>George Miller</director>
    <estreno>15 mayo 2015</estreno>
    <reparto>Tom Hardy</reparto>
    <reparto>Charlize Theron</reparto>
    <reparto>Nicholas Hoult</reparto>
  </pelicula>
  <pelicula categoria="animacion">
    <titulo idioma="ingles">Inside Out</titulo>
    <director>Pete Docter</director>
    <estreno>17 junio 2015</estreno>
    <reparto>Amy Poehler</reparto>
    <reparto>Phyllis Smith</reparto>
    <reparto>Bill Hader</reparto>
  </pelicula>
</cine>
```



- Els documents XML estan formats per arbres d'elements. Cada arbre comença amb un element **root**, del qual van sorgint rames (**branches**) a elements fills. Tots els elements poden tindre elements fills.
- Es sol parlar de **parent**, **child** i **sibling** per determinar el parentesc (**pare**, **fill**, **germà**).
- Tots els elements poden tindre **CONTINGUT TEXTUAL** ("Mad Max") i **ATRIBUTS** (categoria="accion")

• 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"

Generalment, per **analitzar** un fitxer XML hem de **realitzar 2 operacions o comprovacions**:

1. **Correcte**: Consisteix en **comprovar que el document segueix les regles de creació** de documents XML, es a dir, ens hem d'asegurar que el document està "**ben format**".
2. **Vàlid**: Consisteix en comprovar que el document compleix amb les especificacions del nostre vocabulari, es a dir, **que el document XML complisca amb una estructura específica definida en un esquema o document de tipus definit per l'usuari (DTD o XML Schema)**.

• Errors en XML

XML deixa ben clar com s'han de tractar els errors que es produeixen en els documents:

Distingeix entre errors **lleus i greus**, la gestió que ga d'estos errors és diferent

- **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.

• 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.
- **Expat**: Llibreria d'anàlisi XML feta en C
- **Windows.Data.Xml**: Eina de Microsoft gratuïta.

- **Editors XML**

- XML es pot crear amb qualsevol **editor de textos**, alguns fins i tot ofereixen ressaltat de colors com: **Visual Studio Code, Atom, Gedit, Kate, Notepad++**
- Tot i així sempre **ajudarà fer servir algun editor amb suport XML**, ja que ens ofereixen algunes **característiques extres** com:
 - **Auto emplenat**
 - **Validació de documents**
 - **Creació d'esquemes**
 - **Transformacions, etc...**
- **Hi ha editors** com **XML Copy Editor** **específics per** treballar en XM
- Molt sovint, IDEs de **programació tenen suport: IntelliJ, Eclipse, Microsoft Visual Studio, Webstorm.**
- **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.**

2.- XML – Regles Bàsiques

Els documents XML contenen **text**, **etiquetes** i **atributs**

```
<nom carrec="coordinadora"> Isabel S. </nom>
```

- **Etiquetes:** `<nom> </nom>`
- **Atributs:** `carrec="coordinadora"`
- **Text:** `Isabel S.`

No es pressuposa res de com es representaran les dades, l'usuari té el control total sobre com s'organitzen i representen les dades.

- **Regles Bàsiques**

1. Tots els documents tenen una etiqueta arrel
2. Totes les etiquetes s'han de tancar
3. Les etiquetes han d'estar niades correctament
4. Els noms de les etiquetes han de ser correctes
5. Les majúscules i minúscules són diferents
6. Els valors dels atributs han d'estar entre cometes

1.- Etiqueta arrel.

- Tots els documents han de tindre una **etiqueta arrel** que continga totes les altres.
- En el següent exemple seria `<persona>`
- No seria un document XML correcte:

```
<persona>  
  <nom>Aitor Tilla</nom>  
  <nom>Lola Mento</nom>  
</persona>
```

```
<nom>Aitor Tilla</nom>  
<nom>Lola Mento</nom>
```

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

2.- 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" />
```

3.- Correctament niades ("anidadas")

<ul style="list-style-type: none"> Si es comencen dos etiquetes aquestes s'han de tancar correctament 	<ul style="list-style-type: none"> No es poden tancar de qualsevol forma (al estar tancat en ordre invers és incorrecte)
<pre><inventari> <article> Disquet </article> </inventari></pre>	<pre><inventari> <article> Disquet </inventari> </article></pre>

4.- Noms Correctes

Els noms de les etiquetes han de ser correctes, i s'ha de seguir unes regles:

- 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' estiga com estiga escrita.
- No hi pot haver cap espai després de l'inici de la etiqueta '<'.

Correctes	Incorrectes
<Ciutat> </Ciutat>	<XmlElement> </XmlElement>
<Item12> </Item12>	<l0sset> </l0sset>
<casa-Nova> </casa-Nova>	< Item> </ Item>
<element > </element >	<Aitor Tilla> </Aitor Tilla>
<_Hola> </_Hola>	<\$Diners> </\$Diners>

5.- Case Sensitive

Per XML les majúscules són lletres diferents que les minúscules

Correctes	Incorrectes
<pre><Article> Disquet </Article> <article> Disquet </article></pre> <p>(Són 2 etiquetes diferents)</p>	<pre><Article> Disquet </article></pre>

6.- 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 podem tindre atributs sense valor.

Nota: No importa si són cometes simples o dobles

3.- XML – Creació de Documents

• Declaració

La declaració d'un arxiu XML és una línia especial que es col·loca a l'inici del document XML per identificar-lo i proporcionar informació sobre la seva versió i codificació de caràcters. La declaració XML té un format específic i normalment s'assembla a això:

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

La declaració no és obligatòria però sí recomanable. En molts casos, la declaració XML només contindrà els atributs **version** i **encoding**, ja que **standalone** no és necessari en la majoria dels documents XML.

- **<?xml**: marca d'inici de la declaració XML i indica que es tracta d'un document XML.
- **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.
- **encoding**: Defineix el codi de caràcters que fem servir en el document, ja 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:

- **UTF-8**: Aquest és un dels valors més utilitzats i representa la codificació Unicode (UTF-8). És capaç de representar una àmplia gamma de caràcters de diferents idiomes i és eficient en termes d'espai de emmagatzematge.
- **UTF-16**: Aquest valor indica que s'utilitza la codificació Unicode (UTF-16). És similar a UTF-8 però utilitza dues unitats de caràcter per representar cada caràcter.
- **ISO-8859-1**: Aquest és un exemple de codificació ISO-8859, que s'utilitza principalment per a idiomes europeus occidentals. També és coneguda com a "Latin-1."
- **ISO-8859-2**: Una altra codificació ISO-8859, utilitzada per idiomes d'Europa central i oriental.
- **ISO-8859-15**: Similar a ISO-8859-1, però amb algunes modificacions per suportar millor els idiomes europeus.
- **Windows-1252**: Una codificació específica de Microsoft que és similar a ISO-8859-1 però amb alguns canvis.
- Altres codificacions com "UTF-32," "ISO-8859-7," "Shift_JIS," i moltes altres, depenent de les necessitats específiques.

- **standalone**. Aquest atribut pot tenir els valors "yes" o "no" i indica si el document XML és independent o no d'altres recursos externs. (Si necessita un altre fitxer o no)
 - **standalone="yes"**: Significa que el document està complet i per tant no li calen altres documents
 - **standalone="no"**: Vol dir que el document no es pot entendre per si soles. (Generalment degut a que li falta algun fitxer per validar-lo)

Per tant, la declaració:

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

Ens informa de que el fitxer XML

- Segueix XML versió 1.0
- Utilitza Caràcters Unicode
- Necessita un DTD (Document Type Definition) per ser validat

• Etiquetes

- Sempre van entre els símbols "<" i ">"
- Les etiquetes són creades per l'autor i han de ser autoexplicatives, es a dir, han de descriure les dades que contenen.
- Dins d'una etiqueta podem tindre tants atributs com necessitem

Per exemple:

```
<poble> La Vall d'Uixó </poble>
```

(Deixem clar que és un poble)

• Atributs

En XML, els atributs són elements de metadades que es poden afegir a les etiquetes per proporcionar informació addicional sobre els elements o per controlar el comportament de com es processa o es presenta el contingut de l'element.

```
<nom carrec="coordinadora">
```

Els atributs són parells de noms i valors que es col·loquen dins d'una etiqueta XML per proporcionar informació addicional sobre l'element que etiqueten

- Els atributs són un parell format per clau=valor
- El nom dels atributs ha de ser un nom XML vàlid i no repetit (ha de complir les regles)
- Podem crear tants atributs com necessitem
- El valor de l'atribut pot estar entre cometes simples (') o dobles (")
- **OJO:** Tots els atributs han de tindre alguna cosa assignada

Per exemple.

<nom carrec=" ">	->	CORRECTE !!
<nom carrec>	->	INCORRECTE !!

Els atributs **s'utilitzen** per afegir **metadades** a elements XML o per controlar com es processa i es presenta el contingut de l'element. Per exemple:

```
<persona nom="Lola" cognom="Mento" edat="30" />
```

- **Metadades:** Poden proporcionar informació sobre un element, com ara el nom i el cognom d'una persona en l'exemple anterior.
- **Control del contingut:** Poden influir en com **es processa o es presenta** el contingut de l'element. Per exemple, un atribut **"href"** en una etiqueta **<enllaç>** pot especificar la URL d'un enllaç web.
- **Idioma i codificació:** S'utilitzen per especificar **l'idioma o la codificació** de caràcters d'un document XML.
- **Control d'estil:** En XML utilitzat per documents XHTML o HTML, els atributs com **"style"** o **"class"** s'utilitzen per aplicar estils CSS o classes CSS a elements HTML.

En el següent, l'element **<cotxe>** té tres atributs: **"marca," "model"** i **"any,"** que proporcionen informació sobre el cotxe.

```
<cotxe marca="Toyota" model="Camry" any="2023" />
```

En XML existeixen **atributs predefinits**. Estos atributs tenen un significat especial i i serveixen per controlar diversos aspectes de la representació i el processament del document XML i que poden ser utilitzades per processadors XML o altres eines.

Alguns d'ells són:

- **xmlns:** Utilitzat per definir l'espai de noms (namespace) d'un element o d'un conjunt d'elements dins del document XML.
- **xmlns:xsi:** Defineix l'espai de noms XML Schema Instance i s'utilitza per a la validació d'esquemes XML.
- **xsi:schemaLocation:** Conté la ubicació de l'esquema XML utilitzat per validar el document XML.
- **xml:lang:** Especifica l'idioma principal del contingut de l'element.
- **xml:id:** S'utilitza per assignar un identificador únic a un element.
- **xml:base:** Especifica una URL base que es pot utilitzar per resoldre les URL relatives contingudes a l'element o el document XML.
- **xml:space:** Utilitzat per controlar l'espai en blanc (whitespace) a l'interior del contingut d'un element.
- **xml:version:** Indica la versió de XML utilitzada en el document. Per exemple, 1.0 o 1.1.
- **xml:encoding:** Indica la codificació de caràcters utilitzada en el document XML.
- **xml:standalone:** S'utilitza per especificar si el document XML està en mode "standalone" o si depèn d'entitats externes.
- **xml:comment:** Aquest no és un atribut estàndard, però s'utilitza per incloure comentaris dins del codi XML.

Per exemple:

- **xml:lang**: Permet especificar en quin idioma està el contingut. Ajuda els processadors XML a determinar l'idioma correcte per a la presentació o el processament del 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, té dos valors possibles: "default" i "preserve".

```
<paragraf xml:space="default">
    Aquest és un exemple    amb espais en    blanc.
</paragraf>
```

Amb **xml:space="default"**, els espais en blanc innecessaris a l'interior del contingut de l'element s'eliminen.

Amb **xml:space="preserve"**, els espais en blanc es mantenen.

• Entitats

En XML podem posar qualsevol tipus de text, no te restriccions sobre caràcters, ni sobre el tamany del text ni tampoc restriccions de idioma ...

... excepte uns quants caràcters especials com <, >, &, o caràcters no imprimibles com **tabuladors** o **salts de línea**.

Les **entitats** en XML són mecanismes que permeten representar i referenciar estos caràcters especials o seqüències de caràcters.

Les entitats més comunes són:

Simbol	Transcripció
<	<
>	>
"	"
'	'
&	&

Per Exemple:

```
<paragraf>El &lt; tonto&gt; del poble.</paragraf>
```

Quan es processem el document, el text es mostrarà amb els caràcters < i > correctament representats.

Nota: Si no volem canviar els caràcters per entitats podem usar blocs CDATA

• CDATA

- Un bloc CDATA (acrònim de "Caràcters de Dades") és una **secció especial** dins d'un document XML que **permet incloure text o dades** que contenen caràcters especials
- És útil quan vols incrustar fragments de text que es podrien confondre amb marques XML o contingut que no vols que sigui analitzat o interpretat com a part de l'estructura del document XML.

Un bloc CDATA es defineix amb l'ús de la següent sintaxi:

```
<![CDATA[ Text o dades que poden contenir caràcters especials ]]>
```

Per exemple:

```
<descripcio>
  <![CDATA[
    <p>Aquesta és una pàgina d'exemple.</p>
    <a href="http://www.exemple.com">Visita el nostre lloc web</a>
  ]]>
</descripcio>
```

En este exemple, el contingut dins del bloc CDATA (entre <![CDATA[i]]>) es tracta com a dades i no com a marques XML.

Això significa que tot el text HTML, incloses les etiquetes <p> i <a>, es conserva sense cap interpretació com a marques XML.

Pot ser útil quan volem emmagatzemar fragments de codi HTML, codi font (Javascript), o qualsevol altre text que tinga caràcters que serien interpretats de manera diferent en XML.

• Comentaris

Es poden afegir comentaris en els documents XML, estos comentaris seran ignorats pels programes que processen 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 són parts del document XML que **permeten que programes externs realitzen tasques específiques**.

Estes instruccions estan **delimitades per les marques <? i ?>**, i el primer element dins d'aquestes marques indica quin programa o aplicació ha de processar esta instrucció.

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

On:

- **nom_del_processador** és el nom del programa o servei que ha de processar aquesta instrucció.
- **funcio(valor)** és una tasca específica que s'ha de realitzar amb un valor determinat.

Exemple:


```
<?calculadora suma(5, 7) ?>
```

• Exemple de com associar un arxiu CSS a un document XML.

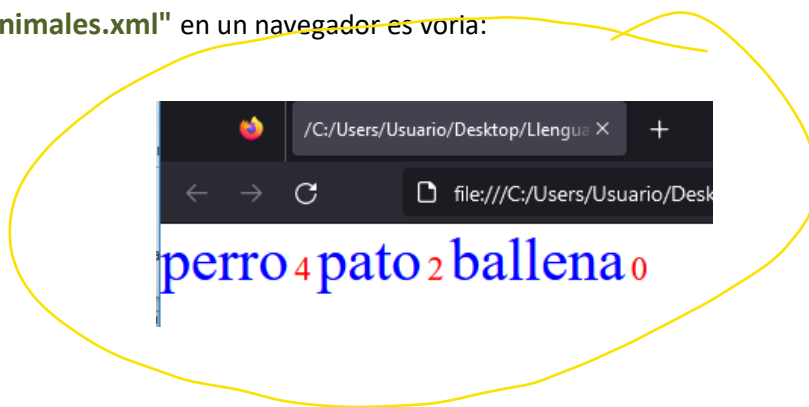
En un document XML podríem escriure la següent instrucció de processament:

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

Aquesta instrucció serveix per a associar l'arxiu CSS (Cascading Style Sheets) "estilo-animales.css" al document XML. Aquest arxiu podria contindre, per exemple, el següent codi:

"estilo-animales.css"	"animales.xml"
<pre>nombre{color:blue;font-size:40px} patas{color:red;font-size:22px}</pre> 	<pre><?xml version="1.0" encoding="UTF-8" ?> <?xml-stylesheet type="text/css" href="estilo- 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></pre>

Al executar "**animales.xml**" en un navegador es voria:



• Espais de Noms (Namespace)

Una de les característiques de XML és que cada persona pot crear tantes etiquetes com vullga amb el nom que vullga. Per tant, al combinar documents XML, podem trobar-nos etiquetes repetides.

Estes etiquetes repetides, encara que tinguen el mateix nom, poden significar coses diferents, i per tant provocar un conflicte que no podríem resoldre.

Per poder solucionar este conflicte hem de trobar alguna solució per poder diferenciar-les, com els **Espais de noms**.

L'objectiu dels **Espais de Noms** és assegurar que els **noms d'elements i atributs siguen únics i no entren en conflicte** quan es combinen diferents documents XML o es creen documents XML complexos amb molts elements. Permeten organitzar les dades de manera jeràrquica i evitar confusions.

Es declaren mitjançant l'ús de l'atribut **xmlns** (XML Namespace) en els elements del document XML.

Aquest atribut s'afegeix a l'element **arrel** del document i pot tenir un valor que és una URI (Uniform Resource Identifier) o una URL.

• La sintaxi bàsica seria:

```
<element_arrel xmlns="http://www.example.com/espai_de_noms">
  <!-- Elements i atributs dins d'aquest espai de noms -->
</element_arrel>
```

- També es pot fer servir una sintaxi amb **prefixos** per associar elements i atributs a espais de noms **específics**. Aquesta forma és útil quan es volen utilitzar diversos espais de noms en el mateix document XML.

Sintaxi:

```
<element_arrel xmlns:prefix="http://www.example.com/espai_de_noms">
  <prefix:element>Contingut</prefix:element>
</element_arrel>
```

Exemple amb **prefixe**:

```
<alumnes:nom>Aitor Tilla</alumnes:nom>
<professors:nom>Lola Mento</professors:nom>
```

Així queda clar a quin espai pertany cadascuna de les etiquetes **nom**.

- 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> Aitor Tilla </A:nom>
```

```
<P:nom> Lola Mento </P:nom>
</classe>
```

- En esta definició trobem:

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

On

- **xmlns:** Atribut per crear espais de nom.
- **A:** Prefixe o alias de l'espai de noms
- **http://server/alumnes** : Nom del *namespace*
- També podem definir quin és l'espai de noms per defecte sense posar-li prefix.


```
<classe xmlns="http://server/alumnes">
...
</classe>
```
- No cal definir tots els espais de noms en l'arrel, també es poden definir directament en els elements, això també és vàlid pels seus descendents

```
<classe xmlns="http://server/alumnes">
<nom>Aitor Tilla</nom>
<nom xmlns="http://server/professors">
  Lola Mento
</nom>
</classe>
```

- Per tant, estes tres versions són equivalents:

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

<classe xmlns:A="http://server/alumnes"
  xmlns:P="http://server/professors">
  <A:nom> Aitor Tilla </A:nom>
  <P:nom> Lola Mento </P:nom>
</classe>
```

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

<classe xmlns="http://server/alumnes"
  xmlns:P="http://server/professors">
  <nom> Aitor Tilla </nom>
```



```
<P:nom> Lola Mento </P:nom>
</classe>
```

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

```
<classe xmlns="http://server/alumnes">
  <nom> Aitor Tilla </nom>
  <nom xmlns="http://server/professors"> Lola Mento </nom>
</classe>
```

En Resum...

Amb els espais de nom podem integra les etiquetes de diferents llenguatges dins d'un document XML (SVG, XHTML, SMIL, ...)

Nom	URL ESPAI DE NOMS
XML	http://www.w3.org/XML/1998/namespaces
XSL	http://www.w3.org/1999/XSL/Transform
XSL-FO	http://www.w3.org/1999/XSL/Format
XHTML	http://www.w3.org/1999/xhtml
SVG	http://www.w3.org/TR/svg