

M4-UF1-NF3: EDICIÓ I FORMAT DE DOCUMENTS XML

1. LLENGUATGES DE MARQUES

Els llenguatges de marques estan per tots els costats, en el nostre ordinador existeixen nombrosos arxius que mai imaginàriem que estan escrits amb un llenguatge de marques.

1.1. Què és un llenguatge de marques?

Un llenguatge de marques és un llenguatge en el qual les seves parts es diferencien entre sí mitjançant senyals. 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 popular és HTML.

El llenguatge extensible de marques XML (eXtensible Markup Language) és un metallenguatge, és a dir, un llenguatge que permet descriure altres llenguatges. XML és més aviat un format estàndard per a l'estructuració de dades que no pas un llenguatge de programació.

XML ofereix la possibilitat de definir etiquetes (*tags*) i les relacions estructurals entre elles.

Com a resultat, els dissenyadors poden crear etiquetes específiques segons les seves necessitats, per tal de definir, intercanviar i validar informació entre múltiples sistemes i aplicacions.

Exemple xml:

```
<?xml version=" 1.0 " encoding=" UTF-8 " standalone=" yes ">
<fitxa>
  <nom>Gabriel</nom>
  <cognom>Molina</cognom>
  <adreça> Colon 36 </adreça>
</fitxa>
```

1.2. Classificació

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			1 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

Existeixen tres classes de llenguatges de marques:

- **Marques de presentació:** aquestes marques indiquen el format-marc del text. El seu ús comença a reduir-se atès que és poc flexible, especialment en grans projectes.
- **Marques de procediments:** aquestes marques s'utilitzen per a la presentació del text, interpretant-se cadascuna en l'ordre que apareixen. Per exemple, la marca que s'agrega immediatament abans d'un text perquè es vegi en negreta. Després ha d'existir la marca corresponent que acabi o tancament la negreta. Altres marques de procediments poden ser centrar text, canvi de grandària de font, canvis d'estils, etc. Alguns llenguatges de marques de procediment són nroff, troff, TeX, PostScript HTML, etc.
- **Marques descriptives:** També cridades marcat descriptiu, o semàntic. Aquí s'utilitzen les marques per a descriure fragments de text sense especificar com han de representar-se. Alguns llenguatges dissenyats per a això són el SGML i el XML. En els llenguatges de marques descriptives el format està separat del contingut, permetent flexibilitat a l'hora de reformatejar un text.

1.3. Exemples de llenguatges de marques

- Darwin Information Typing Architecture (DITA)
- DocBook
- Extensible HyperText Markup Language (XHTML)
- Extensible Markup Language (XML)
- Standard Generalized Markup Language (SGML)
- HyperText Markup Language (HTML)
- Lilypond (sistema per a notació musical)
- Maker Interchange Format (MIF)
- Mathematics Markup Language (MathML)
- Microsoft Assistance Markup Language (MAML)
- Music Extensible Markup Language (MusicXML)
- Rich Text Format (RTF)
- S1000D (Especificació internacional per a documentació tècnica relacionada amb l'àrea comercial i militar).
- TeX, LaTeX (utilitzat generalment en matemàtiques i publicacions acadèmiques).
- Text Encoding Initiative (TEI). (format XML per publicacions digitals)
- Wireless Markup Language (WML), Wireless TV Markup Language (WTVML)

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			2 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

- XHTML Basic (subconjunt de XHTML per a dispositius portàtils, per reemplaçar a WML, XHTML MP i C-HTML).

1.3.1. SGML (Standard Generalized Markup Language)

És un estàndard internacional publicat per la ISO (Organització Internacional d'Estàndards).

SGML va establir dues regles principals:

- La sintaxi que havia d'utilitzar-se per a dissenyar un conjunt de marques aplicables a cada tipus de document. **(ben format)**
- La forma en la qual s'han d'intercalar marques en el text d'un document per a identificar les seves parts estructurals. **(Validació)**

El conjunt de marques que es poden utilitzar amb cada tipus de document constituïx una DTD o definició de tipus de document. El concepte de DTD s'ha reutilitzat també en el llenguatge XML.

1.3.2. HTML (Hyper Text Markup Language)

És una aplicació del llenguatge SGML que especifica com s'han de codificar els documents per a distribuir-los en la Web. El seu origen es remunta a començament dels anys noranta, quan Tim Berners Llig, del CERN, van desenvolupar el World Wide Web.

HTML era independent de plataformes maquinari o programari específiques, el que li convertia en la solució idònia per als problemes d'intercanvi de documentació en format electrònic.

HTML presenta algunes limitacions:

- Incapacitat per a presentar les característiques tipogràfiques i presentacions complexes dels documents.
- La falta de capacitat expressiva del llenguatge, degut al fet que només es pot utilitzar un nombre limitat de marques predefinides en l'especificació.

1.3.3. XML (eXtensible Markup Language)

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			3 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

Va començar a desenvolupar-se en 1996 pel W3C (el comitè encarregat de normalitzar i controlar el desenvolupament dels estàndards per a la Web) amb un clar propòsit: dissenyar un llenguatge de marques optimitzat per a poder ser utilitzat en Internet. Havia de combinar la simplicitat d'HTML, amb la capacitat expressiva de SGML. XML és un llenguatge per a representar dades i informació. XML es diu extensible perquè podem crear les nostres pròpies etiquetes, en lloc d'estar subjectes a un conjunt d'elles com ocorre en HTML.

1.3.4. XHTML (eXtensible Hyper Text Markup Language)

Es presenta com una redefinició del llenguatge HTML utilitzant la sintaxi de XML. El W3C ho va publicar en l'any 1999 i és l'última versió del llenguatge HTML, després de la versió 4.0.

S'han inclòs totes les etiquetes HTML però seguint les directrius de XML. És a dir que , entre altres coses, cada etiqueta que s'obri ha de tancar-se amb un ordre.

2. XML: estructura i sintaxi:

Un aspecte bàsic de XML, és que no es preocupa de la presentació, sinó que parteix de la idea que l'important és **el contingut de les dades** i no la manera que es visualitzaran. Això és ideal per presentar la informació en un format de presentació específic com HTML, PDF, PostScript, etc. A més ens aporta una característica important en informàtica, **tenir les dades separades de la manera de representar-les**.

La base de XML són els **elements**. Un element normalment estarà format per l'obertura d'una **etiqueta, amb atributs o sense, un contingut**, que també potser un grup d'etiquetes i **el tancament de l'etiqueta**.

`<nom>`Joan Reverter`</nom>`

2.1. Parts d'un document XML

Un document XML pot presentar tres parts diferents: el pròleg, el cos i l'epíleg.

2.1.1. PRÒLEG

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			4 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

És una part opcional.

- **Instruccions de processament**

XML proporciona la instrucció de processament com a mitjà alternatiu per passar informació a determinades aplicacions que poden llegir el document.

- Una instrucció de processament comença amb `<?` i acaba amb `?>`
- Les instruccions de processament són marques no elements. Per tant, poden aparèixer en qualsevol part del document XML fora d'una etiqueta.

La instrucció de processament més comú, **xml-stylesheet**, es fa servir per afegir **fulls d'estil** als documents. Apareix sempre abans de l'element arrel.

Exemple:

```
<?xml-stylesheet href="persona.css" type="text/css" ?>
<persona>
    Carla Gómez
</persona>
```

- **Declaració XML**

Es aconsellable començar els documents XML amb una declaració XML. Aquesta declaració sembla a una instrucció de processament però simplement és una declaració XML.

Exemple:

```
<?xml version="1.0" encoding="ASCII" standalone="yes" ?>
<persona>
    Carla Gómez
</persona>
```

Aquesta declaració ha de ser la primera línia del document, no pot estar precedida per cap comentari, espai en blanc, etc.

- **Atribut versió:** L'atribut versió ha de tenir el valor 1.0. En un cas força inusual, també pot tenir el valor 1.1, però aquesta versió limita el document a les versions més recents dels últims analitzadors; però com que tots els analitzadors 1.1 també admeten XML 1.0 no és necessari establir la versió en 1.1.

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			5 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

- **Atribut encoding:** Identifica el joc de caràcters utilitzat per codificar les dades.

Hem comentat que els documents XML es componen de text però hem dit quina codificació fa servir aquest text. Per defecte, els documents XML es suposen codificats en UTF-8 del conjunt de caràcters Unicode.

La majoria dels processadors XML, especialment els escrits en Java, poden controlar una varietat molt més ampla de conjunts de caràcters.

L'únic que hem de fer és indicar-li a l'analitzador quina codificació de caràcters fa servir el document.

Exemple:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"
?>
<persona>
    Carla Gómez
</persona>
```

El conjunt de caràcters ISO-8859-1 inclou lletres com ö i ç. L'atribut encoding és opcional.

- **Atribut standalone:** Indica al processador XML si un document és independent, si el document XML no referència cap altre fitxer, es pot especificar **standalone = "yes"** i si es basa en informació de fonts externes; és a dir, si depèn de declaracions de marca externes, com una DTD externa es pot especificar **standalone = "no"**.

Com que standalone = "no" és l'opció per defecte, rarament veurem standalone en les declaracions XML.

La primera línia s'encarrega de presentar el tipus de document, la versió de la norma a la qual s'adhereix, la codificació de caràcter (US-ASCII, UTF-8, UTF-7, UCS-2, EUCJP, Big5, ISO-8859-1, ISO-8859-7, etc. i altres característiques.

Exemples de pròlegs:

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			6 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

```
<?xml version="1.0" encoding="UTF-7"?>
<!DOCTYPE mensaje SYSTEM "mensaje.dtd">
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<?xml version="1.0" encoding="Big5"?>
```

Altra instrucció de procés seria:

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

2.1.2. COS

Les dades que un document XML ens ofereix estan en el que es coneix com cos.

És un arbre únic d'elements marcats, amb imbricament estricte.

Exemple:

```
<?xml version="1.0"?>
<movies>
  <movie>
    <title>Casablanca</title>
    <director>Michael Curtiz</director>
    <actors>
      <actor>Humphrey Bogart</actor>
    </actors>
  </movie>
  <movie> ... </movie>
</movies>
```

En el nostre exemple, és tot el que va entre **<movie>** i **</movie>**, aquest element es coneix com **l'arrel de l'element**.

2.1.3. EPÍLEG

Situat a continuació del cos, pot estar compost d'instruccions de procés com les del pròleg, llevat de declaracions xml o de tipus de document.

2.2. Elements

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			7 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

Els elements XML poden tenir contingut (més elements, caràcters, o ambdós alhora), o bé ser elements buits.

Un element amb contingut és, per exemple:

`<nom>Joan Reverter</nom>`

La informació, les dades, es distribueix entre els dos tipus d'etiquetes. Així sabem de manera senzilla i ràpida, quina part del document són dades i quines són estructura.

Per diferenciar entre els diferents elements que componen un document XML hem d'utilitzar etiquetes. Les etiquetes estan compostes per un nom i uns atributs, ha d'existir una d'obertura i altra de tancament i han d'estar correctament situades.

Totes les etiquetes que apareixen en els nostres documents XML han de seguir unes normes molt senzilles:

- Han d'estar delimitades pels símbols `<` i `>`.
- El nom de l'etiqueta ha de començar per qualsevol lletra, un guió baix (`_`) o dos punts (`:`). A partir d'aquest moment, podem utilitzar també el guió (`-`)
- Les majúscules i les minúscules són diferents, `<Persona>` no és el mateix que `<persona>` ni `<PERSONA>`.
- Cada etiqueta d'obertura ha de tenir una de tancament.
- Els noms d'elements, d'atributs i altres noms XML han de seguir unes normes, poden contenir:
 - Qualsevol caràcter alfanumèric: lletres de l'alfabet anglès(A..Z,a..z,0..9)
 - Guió baix o caràcter subratllat: `_`
 - Guió: `-`
 - Punt: `.`
 - Els noms d'elements no poden contenir: qualsevol altre signe de puntuació (cometes, apòstrofs, signes de dòlar, signes d'intercalació, símbols de percentatge o punts i comes).
 - No poden contenir espais en blanc, ni retorn de carro, ni salts de línia.

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			8 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

- Només poden començar per lletres o el caràcter de subratllat.

Les etiquetes també poden estar buides. Un element buit, és el qual no té contingut. Es poden escriure de dues maneres diferents:

`<actor></actor>`

o de forma abreujada:

`<actor/>`

2.3. Comentaris

Un document XML pot contenir anotacions en forma de comentari. Els comentaris no són part del contingut d'informació del document, i poden ser ignorats pels processadors XML. Els comentaris s'escriuen com:

`<!-- ...text del comentari... -->`

El text d'un comentari no pot contenir la seqüència --.

2.4. Atributs

Els atributs proporcionen informació addicional sobre un element. Si has vist HTML ja coneixeràs una mica dels atributs. Per exemple, l'etiqueta HTML per a crear un enllaç a altra pàgina podria ser el següent:

`Enlace a yahoo`

Els elements XML poden tenir atributs. Un atribut s'afegeix a l'etiqueta d'inici d'element i té el format:

`<element nom= "valor"> Contingut </element>`

- XML és molt exigent i obliga que el valor de l'atribut estigui entre "" (dobles) o ' (simples).
- Es poden definir tants atributs com facin falta, només cal tenir en compte que han d'anar separats per un o més espais.
- L'ordre dels atributs no importa.
- No es poden repetir noms d'atribut dins un mateix element.

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			9 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

Exemple 1:

```
<persona sexe="femení">
```

Exemple 2:

```
<persona sexe="femení">
  <nom>Carla</nom>
  <cognom>Gómez</cognom>
</persona>
```

Exemple 3:

```
<persona>
  <sexe>femení</sexe>
  <nom>Carla</nom>
  <cognom>Gómez</cognom>
</persona>
```

En l'exemple 2, el sexe és un atribut. En el exemple 3, el sexe és un element. Els dos exemples ens donen la mateixa informació. No hi ha regles respecte a l'ús d'atributs o d'elements.

- Els atributs són útils a HTML.
- En l'XML és aconsellable evitar-los i utilitzar elements al seu lloc.

2.5. Imbricament d'elements

Arbres XML

Els documents XML formen una estructura de dades que s'anomena arbre.

El contingut dels elements no està limitat a només text; els elements poden contenir altres elements, que al seu torn poden contenir text o altres elements, i així successivament.

Un document XML és un **arbre d'elements**. No hi ha límit per a la profunditat de l'arbre, a més de que els elements poden repetir-se.

A l'element que està dintre d'un altre se'l coneix com **fill**. L'element en el qual aquest està contingut es coneix com **pare**.

Considerem ara un element XML més complicat:

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			10 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

```
<persona>
  <nom>Carla </nom>
  <cognoms>
    <primer_cognom>Gómez</primer_cognom>
    <segon_cognom>Sánchez</segon_cognom>
  </cognoms>
  <professio>programadora</professio>
  <professio>matemàtica</professio>
  <professio>escriptora</professio>
</persona>
```

Pares i fills

El document XML anterior està format per un element persona. Aquest element persona conté cinc elements fills: nom, cognoms i 3 elements professió. L'element cognoms té dos elements fills: primer_cognom i segon_cognom. Es diu que l'element persona és el pare dels elements nom, cognoms i dels elements professió. A vegades, es diu que els elements nom, cognoms i els tres elements professió són germans.

L'element arrel

Tots els documents XML tenen **un element sense pare**. Es tracta del primer element del document i és l'element que conté a la resta dels elements. Aquest tipus d'element és l'element arrel del document.

2.6. Entitats predefinides

En XML 1.0, es defineixen cinc entitats per a representar caràcters especials i que no s'interpretin com marcat pel processador XML. És a dir, que així podem usar el caràcter "<" sense que s'interpreti com el començament d'una etiqueta XML, per exemple.

Les entitats són:

Entitat	Caràcter
&	&
<	<
>	>
'	'
"	"

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			11 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

2.7. Seccions CDATA

Es pot donar el cas que el contingut d'un element sigui codi HTML, o codi amb algun llenguatge de programació. Això obligaria a substituir una gran quantitat de símbols i faria el document més il·legible, com per exemple accents, o codi en llenguatge c...

Tot el que està dintre una secció CDATA no és interpretat per cap programa.

Normalment s'utilitzen per:

- Definir grans fragments de text que requereixen moltes substitucions d'entitats.
- Posar contingut d'algun llenguatge de programació com C, JavaScript, HTML,...

Les seccions CDATA es defineixen com les etiquetes normals:

`<![CDATA[` dades de caràcters que no s'interpreten `]]>`

Exemple sense secció CDATA:

```
<titol>
Rock & Roll
</titol>
```

Exemple amb secció CDATA:

```
<titol>
<![CDATA[ Rock & Roll ]]>
</titol>
```

2.8. Eines d'edició:

Podem fer servir editors XML que faciliten la tasca d'edició.
Els editors XML es poden agrupar en:

- **Editors de text adaptats a XML : jEdit**

Funcionalitat bàsica que ha d'oferir un editor de text per a XML:

- Marcar amb diferents colors els elements .

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			12 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

- Auto-completar quan es comença a escriure un element .
- Incorporar sangria de forma automàtica .
- Validar DTD dins del mateix procés de generació de codi XML .

• Eines d'edició gràfica : XML Copy Editor

Representen el document fent servir estructures en arbre, text amb colors, etc.

Normalment permeten verificar si el document està ben 'formatat' i si es 'vàlid'.

Exemple:

persona.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE persona SYSTEM "persona.dtd">
<?xml-stylesheet href="persona.css" type="text/css"?>

<persona>
  <nom>Carla </nom>
  <cognoms>
    <primercognom>Gómez</primercognom>
    <segoncognom>Sánchez</segoncognom>
  </cognoms>
  <professio>programadora</professio>
  <professio>matemàtica</professio>
  <professio>escriptora</professio>
</persona>
```

persona.css

```
persona {font-family: Verdana, sans-serif; color: blue;font-size: 12pt;}
professio{display:block;}
```

Resultat:

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			13 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

Exemple de pro... Curs: J
 Carla Gómez Sánchez
 programadora
 matemàtica
 escriptora

2.9. Elaboració de documents XML ben formats:

Es diu que un document XML està **ben format** quan compleix les **regles sintàctiques indicades**.

Els processadors XML poden rebutjar qualsevol document que no estigui ben format.

A simple vista, un document XML és una estructura jeràrquica d'etiquetes que organitza els elements.

Tot document XML **comença amb una declaració** que l'identifica com a document de tipus XML. Tot i que no és obligatòria, es considera com **una bona pràctica**.

L'aspecte d'aquesta declaració és: `<?xml version="1.0"?>`. Aquesta declaració pot incloure altres informacions, com ara el joc de caràcters que es fa servir en el document:

`<?xml version="1.0" encoding="UTF-16"?>`

Les etiquetes en un fitxer XML segueixen el format següent:

`<nom_etiqueta> </nom_etiqueta>`

És a dir, sempre hi ha una etiqueta que marca l'inici de l'element, i una altra que marca l'acabament. Les etiquetes poden estar estructurades de forma jeràrquica, és a dir, de manera que unes etiquetes continguin les altres, tal com es mostra en l'exemple següent.

Exemple 1: Estructura jeràrquica d'etiquetes XML

```
<?xml version="1.0"?>
<etiqueta1>
  <etiqueta11>
    <etiqueta111>Descripció de l'element</etiqueta111>
    <etiqueta112>Descripció de l'element</etiqueta112>
  </etiqueta11>
  <etiqueta12>Descripció de l'element</etiqueta12>
</etiqueta1>
```

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			14 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

Per tal que l'estructura jeràrquica sigui consistent, és indispensable que la definició de les etiquetes sigui coherent, és a dir, no és correcte que l'etiqueta d'acabament d'un element s'especifiqui abans que l'etiqueta d'acabament de tots els seus sub-elements.

Exemple2: Jerarquia és incorrecta, perquè l'etiqueta de tancament `</etiqueta11>` apareix abans que l'etiqueta `</etiqueta112>`.

```
<?xml version="1.0"?>
<etiqueta1>
  <etiqueta11>
    <etiqueta111>
    </etiqueta111>
    <etiqueta112>
  </etiqueta11>
  </etiqueta112>
</etiqueta1>
```

En tot document, sempre existeix un **element arrel** a partir del qual s'estructuren tots els altres. En el cas de l'Exemple 1, l'element arrel està identificat per l'etiqueta `etiqueta_1`.

Una particularitat de les etiquetes és que poden contenir atributs que fan referència a l'element delimitat per les etiquetes.

En aquest cas, la sintaxi de l'etiqueta és:

```
<etiqueta atribut_1="valor_1" atribut_n="valor_n">....</etiqueta>
```

De manera general, per tal que un **document XML** sigui considerat **sintàcticament correcte**, ha de respectar una sèrie de **regles**:

1. **Tota etiqueta oberta** ha de tenir una **etiqueta de tancament associada**.
2. L'**organització jeràrquica** d'etiquetes ha de ser perfecta.
3. Un document només pot tenir **un element arrel**.
4. Existeix una **convenció** per anomenar les **etiquetes**:
 - El nom ha de començar per una lletra o pel caràcter “_”
 - Les xifres i signes de puntuació només es poden fer servir a partir del segon caràcter.
 - El nom no ha de tenir espais en blanc.

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			15 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

- El nom no ha de tenir el caràcter “.”
- El nom no pot començar per la cadena “xml”
- El nom s'ha d'especificar tot just després del caràcter “<” que marca l'inici de l'etiqueta

5. Quan una etiqueta conté **atributs**, el **valor** de cadascun d'ells ha d'estar delimitat pel caràcter " "

Quan un document XML respecta totes aquestes regles es diu que el document està ben format, és a dir, que és sintàcticament correcte.

Per acabar aquest apartat, es presenta l'Exemple 3 de document XML ben format.

Exemple3:

```
<?xml version="1.0"?>
<Empresa nom="XXX" data_fundacio="2001-11-15">
  <Delegacio nom="Central Barcelona" empleats="100">Text </Delegacio>
  <Delegacio nom="Central Tarragona" empleats="25">Text </Delegacio>
  <Delegacio nom="Lleida" empleats="13">
    <Empleat>
      <Nom>Joan Arbós</Nom>
      <Carrec>Director delegació</Carrec>
    </Empleat>
  </Delegacio>
</Empresa>
```

2.10. Utilització d'espai de noms en XML:

Els conflictes de noms

En XML, els noms dels elements estan definits pel programador. Sovint, això dóna lloc a un conflicte al barrejar documents XML de diferents aplicacions.

Exemple: Aquest document XML conté informació d'**una taula** HTML:

```
<table>
  <fila>
    <col>Mobles Fusta</col>
    <col>Mobles PVC</col>
  </fila>
</table>
```

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			16 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

Aquest XML conté informació **sobre una taula** (un moble):

```
<table>
  <name>Taula de cafè</name>
  <width>80</width>
  <length>120</length>
</table>
```

Si aquests fragments XML es sumen, hi hauria un conflicte de noms. Ambdós contenen un <table> element, però els elements tenen diferent contingut i significat.

Un analitzador XML no sabrà com manejar aquestes diferències. Es necessari resoldre el conflicte entre elements amb el mateix noms.

La solució del conflicte de noms és utilitzant dos espais de noms diferents.

En XML, els espais de noms tenen dos propòsits:

1. Diferenciar entre elements i atributs XML amb el mateix nom.
2. Agrupar tots els elements i atributs del mateix tipus o funció.

Declaració dels espais de noms

- Un espai de noms **es defineix com una referència URI** (*Uniform Resource Identifier*), que servirà per identificar els elements que pertanyen a aquest espai de noms.

Normalment les referències URI són extenses i per tal d'aplicar al llarg de tots els elements del fitxer, resultaria poc pràctic i poc llegible, el que es fa és **assignar un sinònim curt a l'espai de noms**. Així és més pràctic i llegible.

- **On apunta l'URI? A RES!!** (Potser inventat). Es tracta simplement d'un **identificador**.
- Es tracta de donar un **nom únic**.
- De totes formes molts creadors de documents XML utilitzen el namespace com apuntador a una web real que conté informació sobre el namespace. Per exemple: [HTTP://www.w3.org/TR/html4/](http://www.w3.org/TR/html4/)

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			17 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

- Es pot dir que els **elements tindran un nom compost; el seu nom i el nom de l'espai de noms.**

Nota: Els espais de noms **s'apliquen tant a atributs com a elements.**

```
<espainoms:element1>Text1</espainoms:element1>
```

- Per tal d'**assignar un sinònim** a l'espai de noms es fa mitjançant la etiqueta **xmlns**.

```
<exemple xmlns:ej="http://dominiej.com/rutaej">
  <ej:element1>Text1</ej:element1>
  <ej:element2>Text2</ej:element2>
</exemple>
```

Seguint l'exemple anterior, per tal de **solucionar el conflicte amb l'element table** aplicarem els espais de noms:

Espais de Noms XML - L'atribut xmlns

```
<root>
  <h:table xmlns:h="http://www.w3.org/TR/html4/">
    <h:fila>
      <h:col>Mobles Fusta</h:col>
      <h:col>Mobles PVC</h:col>
    </h:fila>
  </h:table>

  <f:table xmlns:f="http://www.w3schools.com/mobles">
    <f:name>Taula de cafè</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>
</root>
```

En aquest exemple, no hi haurà conflicte, perquè els dos elements **<table>** tenen noms diferents. Pertanyen a diferents espais de noms, per tal són diferents.

En l'exemple anterior, l'atribut **xmlns** de l'etiqueta **<table>** dona als prefixes **h:** i **f:** un espai de noms qualificat.

Quan un espai de noms es defineix per a un element, **xmlns:f="http://www.w3schools.com/mobles"** tots els elements descendents amb el mateix prefix estan associats amb el mateix espai de noms.

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			18 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

- L'espai de noms pot ser declarat en els elements on es fa servir o en l'element arrel del document XML. Com podem veure en els següents exemples:

Exemple taula.xml

Declaració de l'espai de noms on es fa servir

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="taula.css" type="text/css"?>
<root>
  <h:tabla xmlns:h="http://www.w3.org/TR/html4/">
    <h:fila>
      <h:col>Mobles Fusta</h:col>
      <h:col>Mobles PVC</h:col>
    </h:fila>
  </h:tabla>
  <f:tabla xmlns:f="http://www.w3schools.com/mobles">
    <f:name>Taula de cafè</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:tabla>
</root>
```

Declaració de l'espai de noms a l'inici

```
<root>
<xmlns:h="http://www.w3.org/TR/html4/">
<xmlns:f="http://www.w3schools.com/mobles">
  <h:table>
    <h:fila>
      <h:col>Mobles Fusta</h:col>
      <h:col>Mobles PVC</h:col>
    </h:fila>
  </h:table>

  <f:table>
    <f:name>Taula de cafè</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>
</root>
```

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			19 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	

Espais de noms per defecte

La definició d'un espai de noms per defecte per a un element ens **estalvia utilitzar el sinònim de nom d'espai curt en tots els elements secundaris**. Té la següent sintaxi:

xmlns="namespaceURI"

Si s'omet el sinònim, llavors la URL que es proporciona és considerada com l'espai de noms per defecte. Aquest espai de noms per defecte s'aplica només als selectors que no tenen cap nom curt d'espai de noms explícitament declarat.

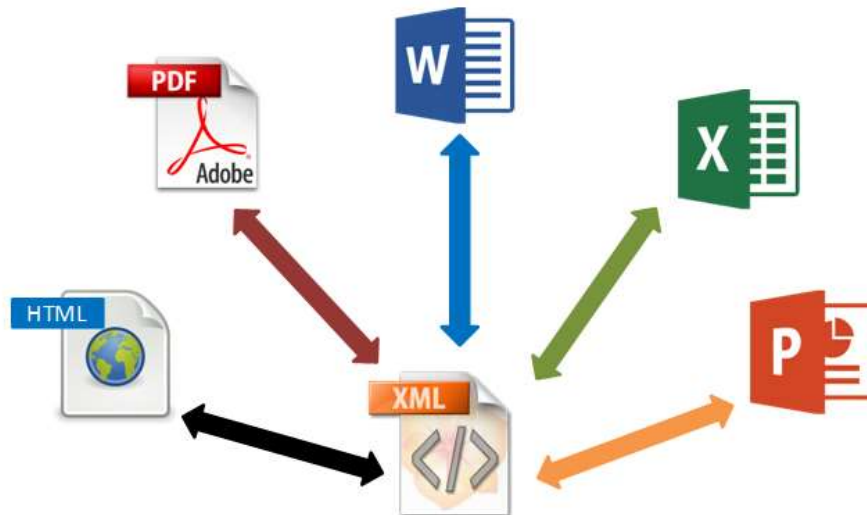
```
<table xmlns="http://www.w3.org/TR/html4/">
  <fila>
    <col>Mobles Fusta</col>
    <col>Mobles PVC</col>
  </fila>
</table>
```

Més informació (<http://www.hipertexto.info/documentos/namespaces.htm>)

USOS DE XML

1. XML com a base de dades
2. Sindicació de notícies
3. XML per guardar gràfics vectorials
4. Actualització de software
5. Formules matemàtiques amb XML
6. Estructures moleculars i informació científica i química.
7. Intercanvi d'informació comptable.
8. Conversor, actualment són molts els formats que ofereixen serveis de conversió a XML com PDF, HTML, DOC

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			20 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	



9. Altres

	Generalitat de Catalunya Departament d'Ensenyament Institut Baix Camp	19/05/2013	Arxiu	Model intern			21 de 21
		versió 3	Elaborat	Cap d'estudis	Cod i	MO-CAP013	