institut sa palomera

ASIX
M-04 Llenguatges de marques

UF1: programació amb XML

A04: validació de fitxers XML

institut sa palomera

Revisió	Data	Autor	Observacions
0	18/10/2022	Josep Bassó	Document inicial
1	02/11/2023	Josep Bassó	Revisió / actualització

institut sa palomera

Continguts

A04. Validació de fitxers XML

3. Definició d'esquemes i vocabularis en XML:

3.1 Utilització de mètodes de definició de documents XML.
3.2 Creació de descripcions.
3.3 Associació amb documents XML.
3.4 Validació.
3.5 Eines de creació i validació.

institut sa palomera

Resultats d'aprenentatge

3. Estableix mecanismes de validació per a documents XML utilitzant mètodes per definir-ne la sintaxi i l'estructura.

Criteris d'avaluació

3.1 Estableix la necessitat de descriure la informació transmesa en els documents XML i les seves regles.
3.2 Identifica les tecnologies relacionades amb la definició de documents XML.
3.3 Analitza l'estructura i la sintaxi específica utilitzada en la descripció.
3.4 Crea descripcions de documents XML.
3.5 Utilitza descripcions en l'elaboració i validació de documents XML.
3.6 Associa les descripcions de documents XML amb els documents XML.
3.7 Utilitza eines específiques de validació.
3.8 Documenta les descripcions de documents XML.

institut sa palomera

institut sa palomera

Índex

1. **Introducció**
    1. Parser
    2. XML Validator
2. XML DTD
3. XML Schema

institut sa palomera

1. Introducció

La definició de l'estructura que haurà de tenir el fitxer és una qüestió de disseny.

Llavors, un fitxer és correcte (ben format) si està d'acord amb les restriccions que ens marca XML.
I és vàlid si està d'acord amb les restriccions que marcarem nosaltres.

Aquestes restriccions s'han d'indicar seguint un format que el programa validador pugui interpretar.

Coses que haurem de definir:

- Les etiquetes, l'ordre en què poden aparèixer i les possibles repeticions.
- Quins atributs poden tenir.
- Quin contingut poden tenir les etiquetes i els atributs.

institut sa palomera

1. Introducció

Llenguatges d'esquemes:

- Document Type Definitions (DTD). L'original / inicial.
- W3C XML definition language (XSD). El que ha esdevingut estàndard.
- Relax NG
- Schematron
- Etc.

El procés per dir si un fitxer XML és correcte i vàlid el fa un tipus de programari anomenat *parser*.

Els *parsers* poden estar com a opcions d'editors, però també com a llibreries per programar.

institut sa palomera

1. Introducció

1.1. Parser

https://www.w3schools.com/xml/xml_parser.asp

Un analitzador sintàtic (*parser*) permet, a partir de l'anàlisi d'una seqüència d'entrada, determinar-ne l'estructura gramatical i comparar-la amb una gramàtica formal.

En el nostre cas llegeix un fitxer XML i ens diu si és correcte o no d'acord a unes normes prefixades, tot creant un objecte (DOM) amb les dades corresponents.

Cal diferenciar entre si un document és:

- **Correcte o ben format**: compleix amb les normes XML.
- O si **és vàlid**: és correcte **i** compleix amb les normes que indicarem nosaltres *ad-hoc*.

institut sa palomera

institut sa palomera

Índex

institut sa palomera

2. XML DTD

https://www.w3schools.com/xml/xml_dtd_intro.asp
https://www.w3schools.com/xml/xml_dtd.asp

Què és un DTD?

Document Type Definition

Defineix l'estructura i els elements i atributs permesos d'un document XML.

Permet que diferents organitzacions acordin un estàndard de fitxer de comunicació.

institut sa palomera

2. XML DTD

2.1. Com posem la definició

DTD en el mateix fitxer

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

<!DOCTYPE note [
<!ENTITY nbsp " ">
<!ENTITY writer "Writer: Donald Duck.">
<!ENTITY copyright "Copyright: W3Schools.">
]>

<note>
<to>Tove</to>   <from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
<footer>&writer; &copyright;</footer>
</note>
```

institut sa palomera

2. XML DTD

2.1. Com posem la definició

DTD en un fitxer apart - privat

fitxer.xml

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

Note.dtd

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

SYSTEM = DTD privat

institut sa palomera

2. XML DTD

2.1. Com posem la definició

DTD en un fitxer apart - públic

fitxer.xml

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

PUBLIC = DTD estàndard
FPI = Formal Public Identifier

institut sa palomera

2. XML DTD

2.1. Com posem la definició

DTD en un fitxer apart amb regles en el propi fitxer

fitxer.xml

<!DOCTYPE note PUBLIC FPI "Note.dtd" [regles] >

<!DOCTYPE note PUBLIC "-//IOC//Classe 1.0//CA" "Note.dtd" [regles] >

institut sa palomera

2. XML DTD

2.1. Com posem la definició

Què composa un fitxer DTD?

Elements: són els blocs principals. Defineixen les etiquetes.

Atributs: defineixen informació extra pels elements.

Entitats: es poden definir caràcters especials o strings a fer servir en el document.

PCDATA: parsed character data. Text entre els tags (etiquetes) que serà parsejat (interpretat). Conseqüència, les entitats han d'estar transformades (&ent;).

CDATA: character data. Text entre els tags que no serà parsejat.

institut sa palomera

2. XML DTD

2.2. Definició d'elements

Declaració d'un element

<!ELEMENT nom_element categoria> o <!ELEMENT nom_element (contingut_element)>

Possibles categories:

EMPTY: no hi haurà informació (<!ELEMENT element-name EMPTY>). Ex.: <!ELEMENT br EMPTY>

PCDATA: <!ELEMENT nom_element (#PCDATA)>

ANY: pot contenir qualsevol combinació de dades parsejables. <!ELEMENT nom_element ANY>

Seqüències:

institut sa palomera

2. XML DTD

2.2. Definició d'elements

Seqüències:

<!ELEMENT nom_element (fill1)> o <!ELEMENT nom_element (fill1, fill2, ...)>

Declaració 1 ocurrència.
Declaració de fills que apareixeran **obligatòriament un sol cop** i **en aquest ordre**.

```
<!ELEMENT note (to, from, heading, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

L'element <note> tindrà 4 fills (to, from, heading i body) que apareixeran obligatòriament 1 sol cop.

institut sa palomera

2. XML DTD

2.2. Definició d'elements

Modificadors:

Declaració de 1 ocurrència: <!ELEMENT nom_element (nom_fill)>

Declaració de 1..N ocurrències, +: <!ELEMENT nom_element (nom_fill+)>

Declaració de 0..N ocurrències, *: <!ELEMENT nom_element (nom_fill*)>

Declaració de 0..1 ocurrències, ?: <!ELEMENT element-name (child-name?)>

Declaració d'un contingut o bé (xor) l'altre, |: <!ELEMENT note (to,from,header,(message|body))>

Declaració de contingut mixt: <!ELEMENT note (#PCDATA|to|from|header|message)*>

2. XML DTD

2.2. Definició d'elements

Contingut barrejat:

Es permet definir contingut i elements.
Obligatòriament en primer lloc ha d'aparèixer un #PCDATA i després els elements que es vulgui.

```
<!ELEMENT carta (#PCDATA|empresa|director|comanda)*>
```

No és gaire bona idea fer-ho servir perquè és difícil de gestionar al fer la verificació.

institut sa palomera

institut sa palomera

2. XML DTD

2.3. Definició d'atributs

Declaració:

<!ATTLIST nom_element nom_atribut tipus_atribut valor_atribut>

Exemple:

DTD: <!ATTLIST payment type CDATA "check">
XML: <payment type="check" />

Si apareix el mateix nom d'atribut per més d'una etiqueta > cal definir-lo per cada etiqueta.

institut sa palomera

2. XML DTD

2.3. Definició d'atributs

Possibles tipus d'atribut:

Type	Description
CDATA	The value is character data
(*en1*|*en2*|..)	The value must be one from an enumerated list
ID	The value is a unique id
IDREF	The value is the id of another element
IDREFS	The value is a list of other ids
NMTOKEN	The value is a valid XML name
NMTOKENS	The value is a list of valid XML names
ENTITY	The value is an entity
ENTITIES	The value is a list of entities
NOTATION	The value is a name of a notation
xml:	The value is a predefined xml value

institut sa palomera

2. XML DTD

2.3. Definició d'atributs

Possibles tipus d'atribut:

Llista de possibles valors (llista enumerada) d'un atribut:

<!ATTLIST nom_element nom_atribut (en1|en2|..) valor_defecte>

DTD: <!ATTLIST pagament tipus (xec|efectiu) "efectiu">

XML vàlid: <pagament tipus ="xec" /> o <pagament tipus ="efectiu" />

institut sa palomera

2. XML DTD

2.3. Definició d'atributs

Possibles valors d'atribut:

Value	Explanation
value	The default value of the attribute
#REQUIRED	The attribute is required
#IMPLIED	The attribute is optional
#FIXED *value*	The attribute value is fixed

institut sa palomera

2. XML DTD

2.3. Definició d'atributs

Possibles valors d'atribut:

Valor per defecte d'un atribut: <!ATTLIST nom_element nom_atribut CDATA "valor_defecte">

DTD:

<!ELEMENT quadrat EMPTY>
<!ATTLIST quadrat width CDATA "0">

XML vàlid:

<quadrat width="100" />
<quadrat /> > width val "0"

Valor requerit d'un atribut: <!ATTLIST nom_element nom_atribut tipus_atribut #REQUIRED>

DTD: <!ATTLIST persona numero CDATA #REQUIRED>

XML vàlid: <persona numero="5677" />

XML invàlid: <persona />

institut sa palomera

2. XML DTD

2.3. Definició d'atributs

Possibles valors d'atribut:

Valor opcional d'un atribut: <!ATTLIST nom_element nom_atribut tipus_atribut #IMPLIED>

DTD: <!ATTLIST contacte fax CDATA #IMPLIED>
XML vàlid: <contacte fax="972-123456" />
XML vàlid: <contacte />

Valor fix d'un atribut: <!ATTLIST nom_element nom_atribut tipus_atribut #FIXED "value">

DTD: <!ATTLIST emissor empresa CDATA #FIXED "Microsoft">
XML vàlid: <emissor empresa ="Microsoft" />
XML invàlid: <emissor empresa ="W3Schools" />

institut sa palomera

institut sa palomera

2. XML DTD

2.4. Ús d'elements vs atributs

La informació pot estar emmagatzemada en elements o en atributs.

```
<persona genere="dona">
  <nom>Marge</nom>
  <cognom>Simpson</cognom>
</persona>
```

```
<persona>
  <genere>dona</genere>
  <nom>Marge</nom>
  <cognom>Simpson</cognom>
</persona>
```

Alguns problemes que pot portar l'ús d'atributs:

- No poden contenir múltiples valors.
- No es poden expandir fàcilment (per canvis futurs).
- No poden descriure estructures (els elements tenen estructura d'arbre).
- Són més difícils de manipular amb un programa.
- No són fàcils de verificar amb un DTD.

institut sa palomera

2. XML DTD

2.4. Ús d'elements vs atributs

Exemple amb elements i amb atributs.

```
<note>
 <date>
      <day>12</day>
      <month>11</month>
      <year>2002</year>
 </date>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

```
<note day="12" month="11" year="2002"
to="Tove" from="Jani" heading="Reminder"
body="Don't forget me this weekend!">
</note>
```

Idea de disseny:
Només fer servir atributs quan sigui per posar-hi un identificador (id).

```
<note id="p502">
 <to>Jani</to>
 <from>Tove</from>
 <heading>Re: Reminder</heading>
 <body>I will not!</body>
</note>
</messages>
```

institut sa palomera

2. XML DTD

2.5. Entitats

Les entitats es fan servir per definir dreceres a caràcters especials.

Definició d'entitat: <!ENTITY nom_entitat "valor_entitat">

DTD:

```
<!ENTITY escriptor "Margaret Hamilton">
<!ENTITY copyright "Copyright W3Schools.">
```

XML:

```
<autor>&escriptor;&copyright;</autor>
```

Definició externa d'una entitat: <!ENTITY nom_entitat SYSTEM "URI/URL">

DTD:

```
<!ENTITY escriptor SYSTEM "https://www.w3schools.com/entities.dtd">
<!ENTITY copyright SYSTEM "https://www.w3schools.com/entities.dtd">
```

XML:

```
<author>&escriptor;&copyright;</author>
```

institut sa palomera

2. XML DTD

Exemples al moodle.

https://www.w3schools.com/xml/xml_dtd_examples.asp

institut sa palomera

institut sa palomera

Índex

1. Introducció
2. XML DTD
3. **XML Schema**
    1. Com posem la definició
    2. Elements simples
    3. Restriccions
    4. Atributs
    5. Elements complexos
    6. Indicadors

institut sa palomera

3. XML Schema

https://www.w3schools.com/xml/schema_intro.asp

XML Schema (.XSD) descriu l'estructura d'un document XML, com un DTD.

XML Schema és més potent que el DTD i és el que es fa servir més actualment.

- Suporta tipus de dades.
- Fa servir sintaxi XML.
- Fa la comunicació més robusta (a l'incloure tipus de dades, p.ex. dates).
- Facilita trobar errors de significat (semàntics) pel validador de software.

institut sa palomera

3. XML Schema

3.1. Com posem la definició

Exemple:

Arxiu XML:

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

Definició DTD:

```
<!ELEMENT note (to, from, heading, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

XML Schema:

```
<?xml version="1.0"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="https://www.w3schools.com"
xmlns="https://www.w3schools.com"
elementFormDefault="qualified">
<xs:element name="note">
 <xs:complexType>
       <xs:sequence>
       <xs:element name="to" type="xs:string"/>
       <xs:element name="from" type="xs:string"/>
       <xs:element name="heading" type="xs:string"/>
       <xs:element name="body" type="xs:string"/>
       </xs:sequence>
 </xs:complexType>
</xs:element>
</xs:schema>
```

institut sa palomera

3. XML Schema

3.1. Com posem la definició

Referència al fitxer de definició:

Definició DTD:

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

<!DOCTYPE note SYSTEM
"https://www.w3schools.com/xml/note.dtd">

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

XML Schema:

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

<note
xmlns="https://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-inst
ance"
xsi:schemaLocation="https://www.w3schools.com/xm
l note.xsd">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

3. XML Schema

3.1. Com posem la definició

L'element <schema>:

És l'element arrel de l'XSD.

Definir un fitxer d'esquema:

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" => namespace tipus dades i elements.
targetNamespace="https://www.w3schools.com"    => namespace dels elements.
xmlns="https://www.w3schools.com"    => namespace per defecte.
elementFormDefault="qualified">    => els elements han d'estar qualificats a l'espai de noms.
...
...
</xs:schema>
```

institut sa palomera

3. XML Schema

3.1. Com posem la definició

L'element <schema>:

Associar un XML a un XSD:

Mitjançant un espai de noms:

<element xmlns:xsi="http://www.w3.org/2001/XMLSchema−instance"
xsi:noNamespaceSchemaLocation="element_definicio.xsd">

3. XML Schema

3.2. Elements simples

Tipus d'elements:

- Simples: sense atributs, només contenen dades.
- Complexos: poden tenir atributs, tenir o no contingut i contenir elements. (normalment l'arrel).

institut sa palomera

3. XML Schema

3.2. Elements simples

Un element simple és el que conté només un “text” (o una dada).
No pot contenir altres elements o atributs.

```
<xs:element name="xxx" type="yyy"/>

<xs:element name="lastname" type="xs:string"/>
<xs:element name="age" type="xs:integer"/>
<xs:element name="dateborn" type="xs:date"/>

<lastname>Refsnes</lastname>
<age>36</age>
<dateborn>1970-03-27</dateborn>
```

institut sa palomera

3. XML Schema

3.2. Elements simples

Tipus més habituals:

- xs:string => cadena de caràcters.
- xs:decimal => valors numèrics
- xs:integer => valors enters
- xs:boolean => 'true' o 'false'
- xs:date => dates amb format AAAA-MM-DD
- xs:time => hores amb format HH:MM:SS
- xs:anyURI => referències a llocs (URLs, camins a llocs, …)

Valors per defecte i fixats:

```
<xs:element name="color" type="xs:string" default="red"/>
<xs:element name="color" type="xs:string" fixed="red"/>
```

institut sa palomera

3. XML Schema

3.3. Restriccions

A un rang de valors:

En un rang de valors [0..120]:

```
<xs:element name="age">
  <xs:simpleType>
        <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="120"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
```

3. XML Schema

3.3. Restriccions

A un conjunt de valors (enumeracions):

```
<xs:element name="car">
  <xs:simpleType>
        <xs:restriction base="xs:string">
        <xs:enumeration value="Audi"/>
        <xs:enumeration value="Golf"/>
        <xs:enumeration value="BMW"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
```

O, fent una definició de tipus:

```
<xs:element name="car" type="carType"/>

<xs:simpleType name="carType">
  <xs:restriction base="xs:string">
        <xs:enumeration value="Audi"/>
        <xs:enumeration value="Golf"/>
        <xs:enumeration value="BMW"/>
  </xs:restriction>
</xs:simpleType>
```

3. XML Schema

3.3. Restriccions

A una sèrie de valors:

```
<xs:element name="letter">
  <xs:simpleType>
        <xs:restriction base="xs:string">
        <xs:pattern value="[a-z]"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
<xs:element name="initials">
  <xs:simpleType>
        <xs:restriction base="xs:string">
        <xs:pattern value="[a-zA-Z][a-zA-Z][a-zA-Z]"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
```

institut sa palomera

3. XML Schema

3.3. Restriccions

A una sèrie de valors - patrons:

```
<xs:element name="letter">
 <xs:simpleType>
      <xs:restriction base="xs:string">
      <xs:pattern value="([a-z])*"/>
      </xs:restriction>
 </xs:simpleType>
</xs:element>
```

```
<xs:element name="letter">
 <xs:simpleType>
      <xs:restriction base="xs:string">
      <xs:pattern value="([a-z][A-Z])+"/>
      </xs:restriction>
 </xs:simpleType>
</xs:element>

<xs:element name="gender">
 <xs:simpleType>
      <xs:restriction base="xs:string">
      <xs:pattern value="male|female"/>
      </xs:restriction>
 </xs:simpleType>
</xs:element>
```

institut sa palomera

3. XML Schema

3.3. Restriccions

Pels espais en blanc:

Es mantindran els espais en blanc:

```
<xs:element name="address">
  <xs:simpleType>
        <xs:restriction base="xs:string">
        <xs:whiteSpace value="preserve"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Canvia els caràcters LF, CR, Tab, espais per espais:

```
<xs:element name="address">
  <xs:simpleType>
        <xs:restriction base="xs:string">
        <xs:whiteSpace value="replace"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Canvia els caràcters LF, CR, Tab, espais per un espai:

```
<xs:element name="address">
  <xs:simpleType>
        <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        </xs:restriction>
  </xs:simpleType>
</xs:element>
```

3. XML Schema

3.3. Restriccions

De longitud:

```
<xs:element name="password">
 <xs:simpleType>
       <xs:restriction base="xs:string">
       <xs:length value="8"/>
       </xs:restriction>
 </xs:simpleType>
</xs:element>
```

```
<xs:element name="password">
 <xs:simpleType>
       <xs:restriction base="xs:string">
       <xs:minLength value="5"/>
       <xs:maxLength value="8"/>
       </xs:restriction>
 </xs:simpleType>
</xs:element>
```

institut sa palomera

3. XML Schema

3.3. Restriccions

Pels tipus de dades:

Constraint	Description
enumeration	Defines a list of acceptable values
fractionDigits	Specifies the maximum number of decimal places allowed. Must be equal to or greater than zero
length	Specifies the exact number of characters or list items allowed. Must be equal to or greater than zero
maxExclusive	Specifies the upper bounds for numeric values (the value must be less than this value)
maxInclusive	Specifies the upper bounds for numeric values (the value must be less than or equal to this value)
maxLength	Specifies the maximum number of characters or list items allowed. Must be equal to or greater than zero
minExclusive	Specifies the lower bounds for numeric values (the value must be greater than this value)
minInclusive	Specifies the lower bounds for numeric values (the value must be greater than or equal to this value)
minLength	Specifies the minimum number of characters or list items allowed. Must be equal to or greater than zero
pattern	Defines the exact sequence of characters that are acceptable
totalDigits	Specifies the exact number of digits allowed. Must be greater than zero
whiteSpace	Specifies how white space (line feeds, tabs, spaces, and carriage returns) is handled

institut sa palomera

institut sa palomera

3. XML Schema

3.4. Atributs

Els atributs només poden estar definits en elements complexos.

```
<xs:attribute name="xxx" type="yyy"/>
```

Els possibles tipus són els mateixos que pels elements.

Exemple de definició i ús:

```
<xs:attribute name="lang" type="xs:string"/>
<lastname lang="EN">Smith</lastname>
```

Valors per defecte, fixats i opcionals / requerits:

```
<xs:attribute name="lang" type="xs:string" default="EN"/>
<xs:attribute name="lang" type="xs:string" fixed="EN"/>
<xs:attribute name="lang" type="xs:string" use="required"/>
```

institut sa palomera

institut sa palomera

3. XML Schema

3.5. Elements complexos

Un element complex és el que conté altres elements i/o atributs:

Tipus d'elements complexos (que poden tenir, o no, atributs):

- Buits.
- Que contenen altres elements.
- Que contenen només text.
- que contenen elements i text.

institut sa palomera

3. XML Schema

3.5. Elements complexos

Exemples:

- Buit amb un atribut: <product pid="1345"/>

- Que conté altres elements:
<employee>
<firstname>John</firstname>
<lastname>Smith</lastname>
</employee>

- Que conté només text amb un atribut: <food type="dessert">Ice cream</food>

- Que conté text i elements:
<description>
It happened on <date lang="norwegian">03.03.99</date> ....
</description>

institut sa palomera

3. XML Schema

3.5. Elements complexos

Definició d'elements complexos (1): directament anomenant l'element.

Exemple:

```
<xs:element name="employee">
  <xs:complexType>
        <xs:sequence>
        <xs:element name="firstname" type="xs:string"/>
        <xs:element name="lastname" type="xs:string"/>
        </xs:sequence>
  </xs:complexType>
</xs:element>
```

3. XML Schema

3.5. Elements complexos

Definició d'elements complexos (2): definint un tipus de dades.

Exemple (amb reaprofitament de la definició):

```
<xs:element name="employee" type="personinfo"/>
<xs:element name="student" type="personinfo"/>
<xs:element name="member" type="personinfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
        <xs:element name="firstname" type="xs:string"/>
        <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

institut sa palomera

3. XML Schema

3.5. Elements complexos

Anem a complicar-ho: ampliar un tipus complex.

```
<xs:element name="employee" type="fullpersoninfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
        <xs:element name="firstname" type="xs:string"/>
        <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
        <xs:extension base="personinfo">
        <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
        </xs:sequence>
        </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

institut sa palomera

3. XML Schema

3.5. Elements complexos

Elements complexos buits (empty).

```
<product prodid="1345" />
```

```
<xs:element name="product">
  <xs:complexType>
        <xs:attribute name="prodid" type="xs:positiveInteger"/>
  </xs:complexType>
</xs:element>
```

institut sa palomera

3. XML Schema

3.5. Elements complexos

Elements complexos amb només elements.

```
<person>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<xs:element name="person">
  <xs:complexType>
        <xs:sequence>
        <xs:element name="firstname" type="xs:string"/>
        <xs:element name="lastname" type="xs:string"/>
        </xs:sequence>
  </xs:complexType>
</xs:element>
```

institut sa palomera

3. XML Schema

3.5. Elements complexos

Elements complexos amb només text.

```
<shoesize country="france">35</shoesize>
```

```
<xs:element name="shoesize">
  <xs:complexType>
        <xs:simpleContent>
        <xs:extension base="xs:integer">
          <xs:attribute name="country" type="xs:string" />
        </xs:extension>
        </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

3. XML Schema

3.5. Elements complexos

Elements complexos amb contingut mixt.

```
<letter>
  Dear Mr. <name>John Smith</name>.
  Your order <orderid>1032</orderid>
  will be shipped on <shipdate>2001-07-13</shipdate>.
</letter>
```

```
<xs:element name="letter">
  <xs:complexType mixed="true">
      <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="orderid" type="xs:positiveInteger"/>
      <xs:element name="shipdate" type="xs:date"/>
      </xs:sequence>
  </xs:complexType>
</xs:element>
```

institut sa palomera

institut sa palomera

3. XML Schema

3.6. Indicadors

Tipus d'indicadors:

D'ordre:

- All
- Choice
- Sequence

D'ocurrències:

- maxOccurs
- minOccurs

De grups:

- D'elements.
- D'atributs.

institut sa palomera

3. XML Schema

3.6. Indicadors

Indicadors d'ordre:

All: els elements fills poden aparèixer en qualsevol ordre i, obligatòriament, un cop.
Choice: especifica que hi haurà un fill o l'altre.
Sequence: els elements fills han d'aparèixer en un ordre específic.

```
<xs:element name="person">
  <xs:complexType>
        <xs:all>    <xs:choice>    <xs:sequence>
        <xs:element name="firstname" type="xs:string"/>
        <xs:element name="lastname" type="xs:string"/>
        </xs:all>
  </xs:complexType>
</xs:element>
```

3. XML Schema

3.6. Indicadors

Indicadors d'ocurrències:

maxOccurs: el màxim de vegades que un element pot aparèixer. Sense límit: "unbounded".
minOccurs: el mínim de vegades que un element pot aparèixer.

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string" maxOccurs="10" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

3. XML Schema

3.6. Indicadors

Indicadors de grup:

D'elements: per definir conjunts d'elements relacionats.

```
<xs:group name="persongroup">
  <xs:sequence>
        <xs:element name="firstname" type="xs:string"/>
        <xs:element name="lastname" type="xs:string"/>
        <xs:element name="birthday" type="xs:date"/>
  </xs:sequence>
</xs:group>
```

```
<xs:element name="person" type="personinfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
        <xs:group ref="persongroup"/>
        <xs:element name="country" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

3. XML Schema

3.6. Indicadors

Indicadors de grup:

D'atributs: per relacionar un grup d'atributs.

```
<xs:attributeGroup name="personattrgroup">
  <xs:attribute name="firstname" type="xs:string"/>
  <xs:attribute name="lastname" type="xs:string"/>
  <xs:attribute name="birthday" type="xs:date"/>
</xs:attributeGroup>

<xs:element name="person">
  <xs:complexType>
        <xs:attributeGroup ref="personattrgroup"/>
  </xs:complexType>
</xs:element>
```

institut sa palomera

3. XML Schema

Exemples al moodle.

https://www.w3schools.com/xml/schema_example.asp

institut sa palomera