

XSD

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

tipus: • xs:string • xs:decimal • xs:integer • xs:boolean • xs:date • xs:time

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

Diferència amb DTD: No fem referència explícita a l'element.

exemples:

```
< 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" / >
```

Definició d'un Element amb Fills: Element Complex

```
<xs:element name="employee">
  _<xs:complexType>
  __<xs:sequence>
  ___<xs:element name="firstname" type="x-
s:string"/>
  ___<xs:element name="lastname" type="x-
s:string"/>
  __ </xs:sequence>
  _</xs:complexType>
</xs:element>
```

Continuació XSD

Continuació XSD (cont)

```
< element ref= " t:B
"/>
```

```
</all>
```

Pot haver-hi combinació d'indicadors.

Restriccions a les dades: en valors numèrics

a un conjunt de valors

longitud de cadena

amb patró

Expressions regulars

Bibliografia:

-> apunts Jordi Vidal

-> http://edutechwiki.unige.ch/en/XML_Schema_tutorial_-_Basics

-> <https://www.w3schools.com/>

XML

Les dades en XML estan pensades per a guardar informació.

Manipular dades tot filtrant, seleccionant i ajustant.

Com transformem: XSLT i Xpath

Com funciona xslt. Primer afegir línia:

```
< xsl:stylesheet xmlns:xsl = "http://www.w-
3.org/1999/XSL/Transform" version= "1.0" >
```

```
arrel: < xsl:template match = "catalog" />
```

Podem afegir taula:

Afegir entrades a la taula:

```
<xsl:for--each select="cd">
  _<tr>
  __<td>
  ___<xsl:value--of select="title"/>
  __</td>
  __<td>
  ___<xsl:value--of select="artist"/>
  __</td>
  _</tr>
</xsl:for--each>
```

XML continuació

Funcions avançades:ordenar, if, switch, afegir atributs, fusionar dos xml.

Ordenar: `< xsl:for-each select="cd" order-by= "+artist" > //ascendent`

Condicional:

```
<xsl:for-each select="catalog/cd">
```

Xpath

Un llenguatge per a descriure un conjunt d'un o diversos elements i atributs.

Selecció de nodes:

nodename Selects all nodes with the name "nodename".

/ Selects from the root node.

// Selects nodes in the document from the current node that match the selection no matter where they are.

. Selects the current node

.. Selects the parent of the current node

@ Selects attributes

Funcions:

number last() // number position() // number
count(node - set) // node - set id(object) //
string local - name(node - set) // string
local - name() // string namespace - uri (node - set) // string namespace - uri () //
string name(node - set) // string name()

www.edankert.com/xpathfunctions.html

[https:// www.mulberrytech.com/quickref/-
functions.pdf](https://www.mulberrytech.com/quickref/functions.pdf)

<http://www.xsltfunctions.com>

Xquery

És un llenguatge amb una estructura similar al SQL-> Sel-leccionar dades / Determinar condicions / Operacions / Mostrar Resultats

Tot el que és pot fer en XPath és pot fer en XQuery.

Variables no són tipificades

```
let $resultat := " Hola món !"
```

```
return $resultat
```

Obrir un document:

```
let $d := doc ( '/Users/palamos/Google  
Drive/M4 - XML/UF2/material/xquery_intro/-  
videos.xml ' )
```

Consultes a Xquery:

```
let $doc:=doc('/Users/palamos/GoogleDriv-  
e/M4-XML/UF2/material/xquery_intro/vid-  
eos.xml')
```

```
for $a in $doc//actors/actor  
where ends-with ($a, 'Lisa')
```

```
return $a
```

Aniament d'Elements Complexes + Atributs

Un cop tanques la seqüència...

```
< xs:attribute name="orderid" type="xs:string" use="required" / >
```

Seqüència:

ordenada element dins la seqüència

repeticions element conté: minOccurs i/o maxOccurs

or elements <complexType>

<choice>

<element ref="t:A" />

<element ref="t:B" />

</choice>

</complexType>

Qualsevol ordre dels Elements

<complexType>

<all>

<element ref="t:A" />

```
<xsl:if test="price > 10">
  <tr>
    <td><xsl:value-of select="title"/></td>
    <td><xsl:value-of select="artist"/></td>
    <td><xsl:value-of select="price"/></td>
  </tr>
</xsl:if>
</xsl:for-each>
```

XML continuació2

Alternatives, choose (=switch)

```
<xsl:for-each select="catalog/cd">
  <tr>
    <td><xsl:value-of select="title"/></td>
    <xsl:choose>
      <xsl:when test="price > 10">
        <td bgcolor="#ff00ff">
          <xsl:value-of select="artist"/></td>
      </xsl:when>
      <xsl:otherwise>
        <td><xsl:value-of select="artist"/></td>
      </xsl:otherwise>
    </xsl:choose>
  </tr>
</xsl:for-each>
```



By **alxndre**

cheatography.com/alxndre/

Not published yet.

Last updated 11th May, 2017.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>