HTML: introducció

- Arquitectura de les aplicacions web:
  - El protocol **HTTP** per transferir documents web del servidor al client.
  - El llenguatge **HTML** de representació de la documentació web, que inclou a més de text, les regles de format i presentació visual.
  - La **URL** (uniform resource locator) per referènciar unívocament documents web en la xarxa.

Contingut

- HTML
  - CSS
- XML
  - SCHEMA
  - XSL

Contingut

- HTML
  - CSS
- XML
  - SCHEMA
  - XSL

HTML: introducció

- Llenguatge per codificar documents de la WWW
  - Sintaxis i semàntica especificada
  - Consorci World Wide Web (W3C) http://www.w3.org

- Característiques:
  - Basat en etiquetes (tags): <tag> … </tag>
  - Codificació d'estructura lògica (no presentació)
  - Enllaços a altres objectes (per valor o "inline"/per ref. a URL)

- Cada vegada més complex,…
  - menor codificació manual -> més estricte
  - format intern d'editors (de html o genérics)
  - Netscape, Macromedia Dreamweaver, MS Office, MS FrontPage…

Conceptes bàsics

- Etiquetes:
  - Delimiten un conjunt de text / dades
  - Dónen semàntica al text que delimiten
  - En general, han de tenir un inici i un final (hi ha excepcions)

  - Inici zona delimitada per una etiqueta: <etiqueta>
  - Final zona delimitada per una etiqueta: </etiqueta>

  - Exemple: <etiqueta> text delimitat </etiqueta>

- Atributs:
  - Completen la semàntica d'una etiqueta

  - Forma: <etiqueta atribut1="valor" atribut2="valor"> text </etiqueta>

HTML: estructura

Etiquetes amb funcions estructurals de document:

- Inici i final del document HTML:
  - <html> i </html>
- Capçalera:
  - <head> i </head>
- Cos:
  - <body> i </body>

Exemple de l'estructura bàsica d'un document:

```
<html>
   <head>
      <title>Document bàsic</title>
   </head>
   <body>
      Un document molt senzill
   </body>
</html>
```

HTML: estructura

<!-->
<
<A>
<ABBREV>
<ACRONYM>
<ADDRESS>
<APPLET>
<AREA>
<AU>
<AUTHOR>
<B>
<BANNER>
<BASE>
<BASEFONT>
<BGSOUND>
<BIG>
<BLINK>
<BLOCKQUOTE>
<BQ>
<BODY>

<CAPTION>
<CENTER>
<CITE>
<CODE>
<COL>
<COLGROUP>
<CREDIT>
<DEL>
<DFN>
<DIR>
<DIV>
<DL>
<DT>
<DD>
<EM>
<EMBED>
<FIG>
<FN>
<FONT>
<FORM>
<FRAME>
<FRAMESET>
<H1>
<H2>
<H3>
<H4>
<H5>
<H6>
<HEAD>
<HR>
<HTML>
<I>
<IFRAME>
<IMG>
<INPUT>
<INS>
<ISINDEX>
<KBD>
<LANG>
<LH>
<LI>
<LINK>
<LISTING>
<MAP>
<MARQUEE>
<MATH>
<MENU>
<META>
<MULTICOL>
<NOBR>
<NOFRAMES>
<NOTE>
<OL>
<OVERLAY>
<P>
<PARAM>
<PERSON>
<PLAINTEXT>
<PRE>
<Q>
<RANGE>
<SAMP>
<SCRIPT>
<SELECT>
<SMALL>
<SPACER>
<SPOT>
<STRIKE>
<STRONG>
<SUB>
<SUP>
<TAB>
<TABLE>
<TBODY>
<TD>
<TEXTAREA>
<TEXTFLOW>
<TFOOT>
<TH>
<THEAD>
<TITLE>
<TR>
<TT>
<U>
<UL>
<VAR>
<WBR>
<XMP>

HTML: estructura

<!-->
<
<A> enlace
<ABBREV>
<ACRONYM>
<ADDRESS>
<APPLET>
<AREA>
<AU>
<AUTHOR>
<B>
<BANNER>
<BASE>
<BASEFONT>
<BGSOUND>
<BIG>
<BLINK>
<BLOCKQUOTE>
<BQ>
<BODY>

<CAPTION>
<CENTER>
<CITE>
<CODE>
<COL>
<COLGROUP>
<CREDIT>
<DEL>
<DFN>
<DIR>
<DIV>
<DL>
<DT>
<DD>
<EM>
<EMBED>
<FIG>
<FN>
<FONT>
<FORM>
<FRAME>
<FRAMESET>
<H1> header 1
<H2>
<H3>
<H4>
<H5>
<H6>
<HEAD>
<HR>
<HTML>
<I>
<IFRAME>
<IMG> imagen
<INPUT>
<INS>
<ISINDEX>
<KBD>
<LANG>
<LH>
<LI> elemento de lista
<LINK>
<LISTING>
<MAP>
<MARQUEE>
<MATH>
<MENU>
<META>
<MULTICOL>
<NOBR>
<NOFRAMES>
<NOTE>
<OL> lista ordenada
<OVERLAY>
<P>
<PARAM>
<PERSON>
<PLAINTEXT>
<PRE>
<Q>
<RANGE>
<SAMP>
<SCRIPT>
<SELECT>
<SMALL>
<SPACER>
<SPOT>
<STRIKE>
<STRONG>
<SUB>
<SUP>
<TAB>
<TABLE>
<TBODY>
<TD>
<TEXTAREA>
<TEXTFLOW>
<TFOOT>
<TH>
<THEAD>
<TITLE>
<TR>
<TT>
<U>
<UL> lista no ordenada
<VAR>
<WBR>
<XMP>

HTML: scripting de client

- Processat del document pel client (validació camps, càlcul valors, …)
- Javascript:
  - Llenguatge d'scripting de client basat en Java
  - Disposa de diferents "dialectes" en funció del client (navegador)
  - Exemple:
    - En fer click en un enllaç s'obre un avís:

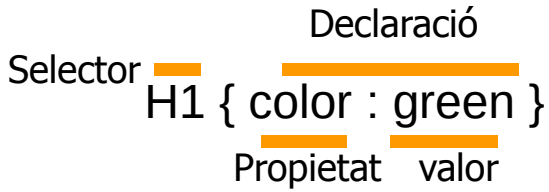
```
<script>
   <a href="pagina.html" onClick="alert('Has fet click');"> Text enllaç </a>
</script>
```

HTML: CSS

- HTML no hauria de fer-se servir per a donar estil a les dades que conté
- La solució correcta és usar fulls d'estil:
  - Cascade Style Sheets (CSS)
- Els fulls d'estil es creen per a controlar l'aparença (en el sentit més visual de la paraula) d'un document HTML:
  - Complementen la info estructural d' HTML
  - Important la separació estructura i estil
- Per què "en cascada"?
  - Info d'estil que es va superposant
  - Van "caient" (aplicant) sobre el document …

HTML: css

- Exemple de regla (mostrar una etiqueta html)



- ¿Què podem controlar?
  - Fonts (color, size, caps, font, etc)
  - Background (image, color, tiling properties)
  - Text (spacing, line-height, alignment, decoration, word-spacing)
  - Box properties (margin, border, float)
  - List properties (image for bullets)
  - Links (visited, hover, active, link)

Contingut

- HTML
  - CSS
- XML
  - SCHEMA
  - XSL

XML

- XML: eXtensible Markup Language
- A partir d’elements <x> … </x>
- XML
  - permet als usuaris crear els seus tags
  - permet separar el contingut de la presentació
  - *Útil per representar informació que pugui ser processable automàticament*

XML: Motivació inicial

- Separació de presentació i dades:

```
<h2>Hamlet, Príncipe de Dinamarca</h2><p>
<b>Autor: </b>William Shakespeare<br>
<b>ISBN: </b>84-239-0027-4<br>
<b>Año: </b>1938</p>
```

HTML **barreja dades i presentació**.

P.ex:

-Nom d'un autor

-Text en negreta

La versió HTML pot ser interpretada per persones mitjançant els browsers, **però els programes NO poden extreure fàcilment** la informació del document HTML

XML: Motivació

- XML estructura les dades i elimina, dins del document, informació sobre la presentació

```
<?xml version="1.0"?>
<libro>
  <autor> <apellido>Shakespeare</apellido>
        <nombre>William</nombre>
  </autor>
  <titulo>Hamlet  Principe de Dinamarca</titulo>
  <isbn>84-239-0027-4</isbn>
  <fecha>1938</fecha>
</libro>
```

- Amb XML, una persona **o un programa** podrà obtenir la informació (apellido, nombre, título, isbn, fecha) amb facilitat

XML: usos habituals

- Usos de XML:
  - Per transportar informació entre bases de dades
  - Per enviar informació que es mostra a l'usuari
  - Com format llegible per expressar dades estructurades
  - Per codificar dades en la xarxa
  - Com a alternativa a formats de dades binàries
- Restriccions:
  - Les dades binàries han de ser enviades com a Base64 o enviar a banda del document XML (un enllaç, com fa HTML per exemple)
  - Força text! (pot comprimir-se)

XML: components

- Per a poder estructurar la informació, XML es basa en els següents components:
  - La **sintaxi** del llenguatge (fixada pel W3C):
    - Cada etiqueta ha d'obrir-se i tancar-se
    - Les etiquetes són "case-sensitive" (majúscules/minúscules)
    - …
  - El **conjunt d'elements** vàlid (fixat per l'usuari):
    - **espai de noms** vàlid per als elements (etiquetes)
      - <libro>, <autor>, <apellido>, <nombre>, <titulo> …
    - les **restriccions d'us**:
      - quan es poden usar i quins valors pode prendre

XML: sintaxi

- Elements poden estar imbricats
  <libro><autor>William S.</autor>…</libro>

- Elements poden tenir atributs
  <libro autor="William S" titulo="Hamlet">

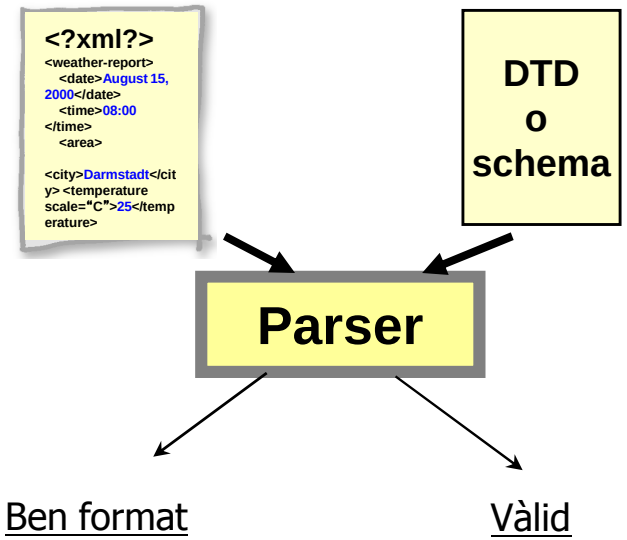
- Els elements s'han de tancar sempre (a HTML no sempre)
  (Incorrecte) <libro><autor>William S.</libro>

- Elements sense dades poden tancar-se al final de la etiqueta
  <libro autor="William S" titulo="Hamlet"/>
  <libro autor="William S" titulo="Hamlet"></libro>

- Codificació de caràcters especials (com a HTML)
  <libro titulo="El "Aleph""/>

XML: validesa

- DEFINICIONS:
  - Document XML “ben format”:
    - Un document que **compleix amb la sintaxi** de xml
  - Document XML “vàlid”:
    - Si a més de ser ben format, **compleix amb un conjunt de restriccions especificades en un fitxer de restriccions**



XML: validesa

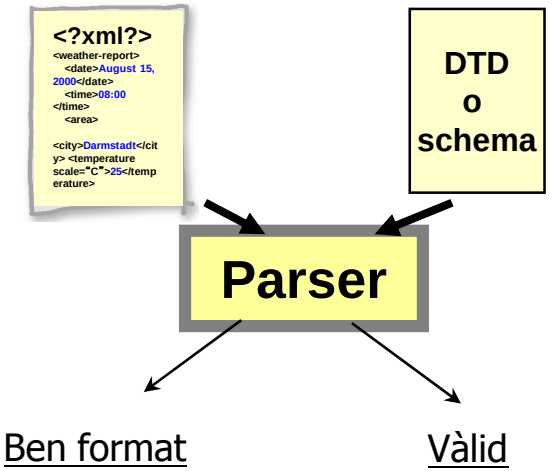
- Els fitxers de restriccions expressen carácterístiques sobre el domini de dades que és representat pel document XML
- Els creen els usuaris o les comunitats
- Expresen gramàtiques que han de complir els elements dels documents XML (a part de la sintaxi genèrica del llenguatge XML)

XML: validesa

- Existeixen diversos formats per als fitxers de restriccions, proposats pel W3C:

1. DTD (Document Type Definition):
   - Definició Tipus Document;
   - 1ª generació: heretat de SGML,
   - sintaxi no XML, sense tipus de dades, …

2. **XML Schema**:
   - format XML,
   - tipus de dades,
   - més restriccions



Contingut

- Antecedents i evolució
- HTML
  - CSS
- XML
  - SCHEMA
  - XSL

Schema

- Funcions:
  - Definir/restringir el contingut i estructura de documents XML usant XML
- Característiques:
  - Facilita mapats amb estructures de dades de programa
  - És un document XML (parsers XML el poden tractar)
- Elimina bona part de les limitacions dels DTDs:
  - Tipus de dades definits per l'usuari
  - Quantificadors (minOccurs, maxOccurs)
  - Refinament (suporta herència de tipus)

Schema: exemple

XML schema

```
<element name='libro' type='TipoLibro'/>
<complexType name='TipoLibro'>
  <sequence>
   <element name='titulo' type='linea'/>
   <element name='autor' type='linea'/>
   <element name='año' type='sigloXX'/>
   <element name='precio' type='decimal'>
    <complexType>
       <attribute name='divisa' type='string'/>
    </complexType>
   </element>
  </sequence>
  <attribute name='isbn' type='ID'/>
 </complexType>
```

DTD

```
<!ELEMENT libro  (título, autor+, año?, precio)>
<!ELEMENT título (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT año (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ATTLIST libro isbn ID #REQUIRED>
```

```
<?XML version="1.0">
<libro isbn="123456">
<título>El Aleph</título>
<autor>J. L. Borges</autor>
<año>1946</año>
<precio divisa="EUR">15</precio>
</libro>
```

XML

Aquí només es declara un tipus (TipoLibro)
S'usen tipus encara no declarats

Schema: exemple

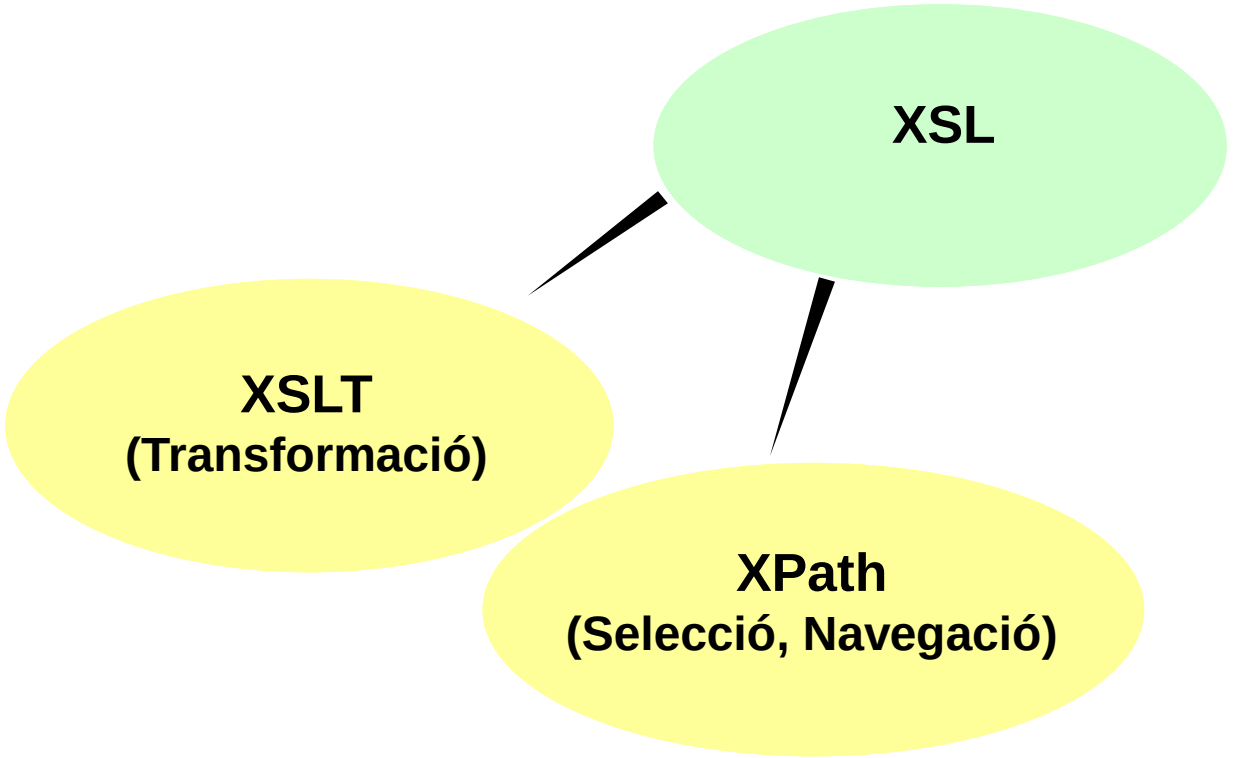
```
<schema>
<simpleType name='linea'>
  <restriction base='string'> <maxLength value='50'/> </restriction>
</simpleType>

<simpleType name='sigloXX'>
  <restriction base='integer'> <minInclusive value='1900'/> <maxInclusive value='1999'/> </restriction>
</simpleType>

<element name='libro' type='TipoLibro'/>

<complexType name='TipoLibro'>
  <sequence>
   <element name='titulo' type='linea'/>
   <element name='autor' type='linea'/>
   <element name='anyo' type='sigloXX'/>
   <element name='precio' type='decimal'>
    <complexType>
        <attribute name='divisa' type='string'/>
    </complexType>
   </element>
  </sequence>
  <attribute name='isbn' type='ID'/>
 </complexType>
</schema>
```

Contingut

- Antecedents i evolució
- HTML
  - CSS
- XML
  - SCHEMA
  - XSL

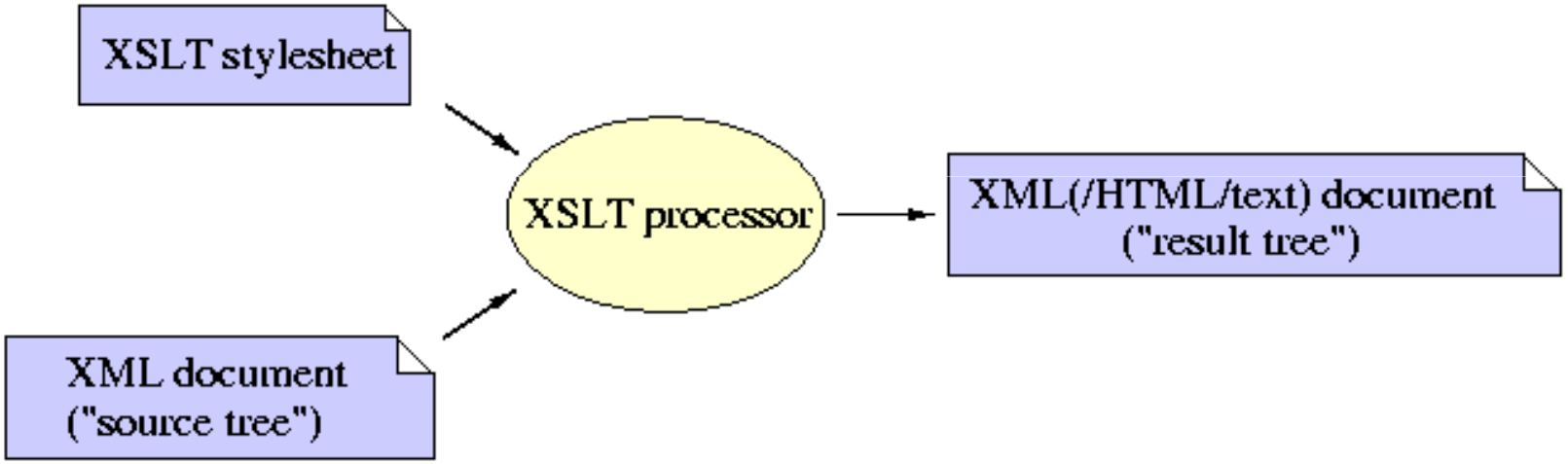
XSL

- XSL: Extensible Stylesheet Language
  http://www.w3.org/Style/XSL



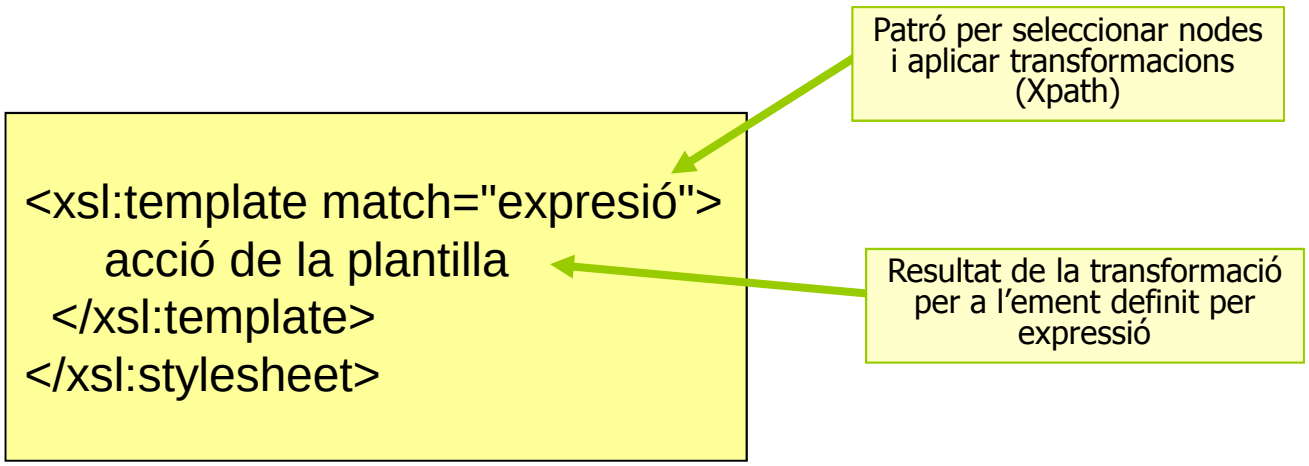
XSLT

- Esquema conceptual de funcionament



XSLT

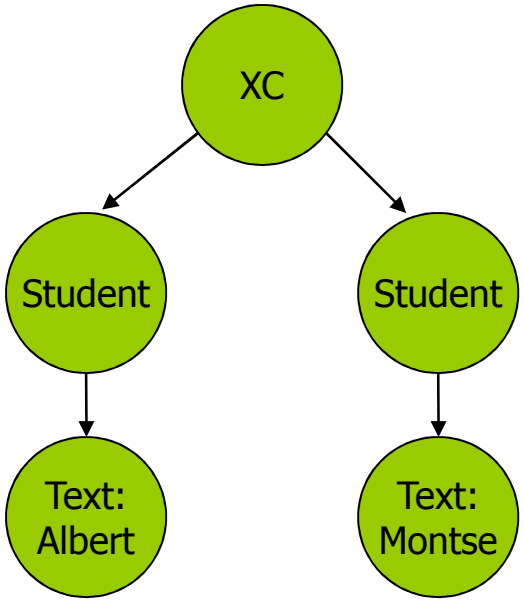
- Procés de transformació de documents:
  - Un conjunt de regles que s'apliquen als elements.
  - Cada regla descriu quina sortida hi haurà per cada element del XML original
    - Cal poder identificar els elements! (Xpath)



XSL: Xpath

- XPath: permet navegar en l'arbre/fitxer XML

```
<XC>
   <Student>Albert</Student>
   <Student>Montse</Student>
</XC>
```



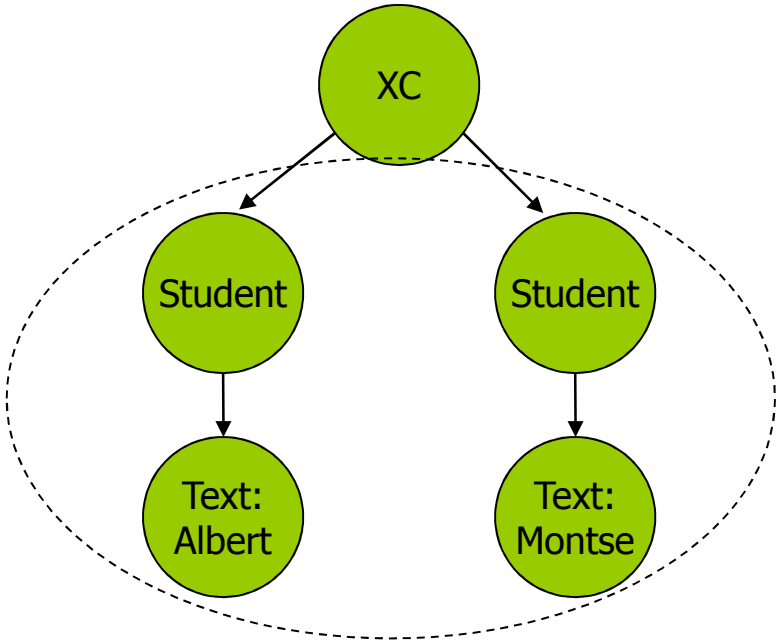
XSL: Xpath

- Exemple:

  /XC/Student

  selecciona tots els estudiants (cadascun d'ells)

```
<XC>
  <Student>Albert</Student>
  <Student>Montse</Student>
</XC>
```



XSL: XSLT

- Definició del patró de transformació:
  - Element <xsl:template>
    - Un full d'estil XSL consisteix en un conjunt de regles anomenades template.
    - Cada element <xsl:template> conté regles per aplicar quan un node determinat fa matching.
    - Atribut match=
      - S'usa per associar el template amb un element XML.
      - També pot ser usat per definir templates per tota una branca d'un document XML.
  - Element <xsl:value-of>
    - Extreu el valor del node seleccionat.
    - L'atribut pot contenir una expressió
  - Element <xsl:for-each>
    - ens permet realitzar bucles en XSL (selecciona cada element XML d'un conjunt de nodes especificat).

XSL: XSLT

- Exemple 1:

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<?xml-stylesheet
type="text/xsl"
href="cdcatalog.xsl"?>

<CATALOG>
  <CD>
    <TITLE>Four Women (4CDs)
    </TITLE>
    <ARTIST>
       Nina Simone
    </ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Verve</COMPANY>
    <PRICE>60.58</PRICE>
    <YEAR>2003</YEAR>
  </CD>
  <CD>
    <TITLE>Music</TITLE>
    <ARTIST> Madonna </ARTIST>
   ….
</CAGTALOG>
```

XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
 <html>
 <body>
  <h2>My CD Collection</h2>
  <table border="1">
   <tr bgcolor="#9acd32">
    <th>Title</th>
    <th>Artist</th>
   </tr>
   <tr>
    <td><xsl:value-of select="catalog/cd/title"/></td>
    <td><xsl:value-of select="catalog/cd/artist"/></td>
   </tr>
  </table>
 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

XSLT

```
<html>
 <body>
   <h2>
      My CD Collection
   </h2>
   <table border="1">
     <tr bgcolor="#9acd32">
       <th>Title</th>
       <th>Artist</th>
     </tr>
     <tr>
      <td>Four Women (4CDs)
      </td>
      <td>Nina Simone</td>
     </tr>
   </table>
 </body>
 </html>
```

HTML

transformació

XSL: XSLT

- Exemple 2 (ús de for-each):

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<?xml-stylesheet
type="text/xsl"
href="cdcatalog.xsl"?>

<CATALOG>
  <CD>
    <TITLE>Four Women (4CDs)
    </TITLE>
    <ARTIST>
       Nina Simone
    </ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Verve</COMPANY>
    <PRICE>60.58</PRICE>
    <YEAR>2003</YEAR>
  </CD>
  <CD>
    <TITLE>Music</TITLE>
    <ARTIST> Madonna </ARTIST>
   ….
</CAGTALOG>
```

XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
 <html>
 <body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Title</th>
      <th>Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
    <tr>
     <td><xsl:value-of select="catalog/cd/title"/></td>
     <td><xsl:value-of select="catalog/cd/artist"/></td>
    </tr>
    </xsl:for-each>
  </table>
 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

XSLT

```
<html>
 <body>
   <h2>
      My CD Collection
   </h2>
   <table border="1">
     <tr bgcolor="#9acd32">
       <th>Title</th>
       <th>Artist</th>
     </tr>
     <tr>
      <td>Four Women (4CDs)
      </td>
      <td>Nina Simone</td>
     </tr>
     <tr>
      <td>Music</td>
      <td>Madonna</td>
     </tr>
   </table>
 </body>
 </html>
```

HTML

transformació

Diagrama relació de llenguatges