XSLT

Tecnólogo en Informática

Ing. Sebastián Vergara
svergara@fing.edu.uy

XSLT

- Formatear XML con CSS no es lo mas común
  - W3C recomienda XSLT

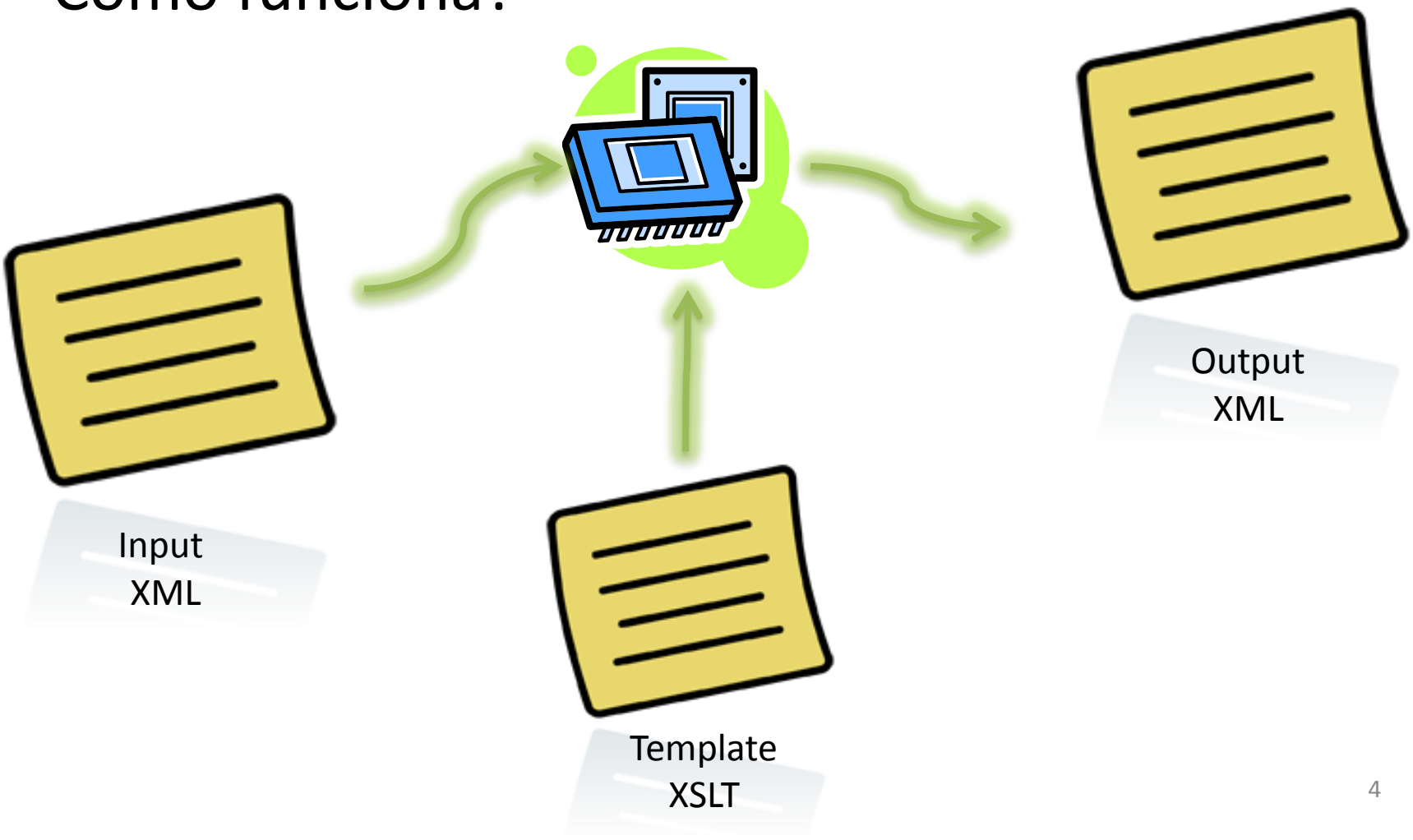
- XSLT? e**X**tensible **S**tylesheet **L**anguage **T**ransformations
  - Parte de XSL
    - Incluye vocabulario para especificar formato
    - Otras especificaciones en XSL
      - Xpath: navegacion de documentos XML
      - XSL-FO: formateo de documentos XML
  - Especifica como se transforma un XML en otro
  - Independiente de XSL

XSLT

- Con XSLT se puede transformar un XML en cualquier otro documento basado en XML
  - HTML
  - XHTML
- Se apoya en
  - XPath
    - Para navegar atributos del documento XML
    - Para navegar elementos del documento XML
    - Encontrar información particular

XSLT

- Cómo funciona?



XSLT

- Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<catalogo>
  <cd>
    <titulo>Empire Burlesque</titulo>
    <artista>Bob Dylan</artista>
  </cd>
  <cd>
    …
  </cd>
  …
  <cd>
    <titulo>1999 Grammy Nominees</titulo>
    <artista>Many</artista>
  </cd>
</catalogo>
```

XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:template match="/">
      <html><body>
         <h2>Mi colección de cds</h2>
         <table border="1">
            <tr bgcolor="#9acd32">
               <th>Titulo</th>
               <th>Artista</th>
            </tr>
            <xsl:for-each select="catalogo/cd">
               <tr>
                  <td><xsl:value-of select="titulo"/></td>
                  <td><xsl:value-of select="artista"/></td>
               </tr>
            </xsl:for-each>
         </table>
      </body></html>
   </xsl:template>
</xsl:stylesheet>
```

XSLT

- Output XML

```
<html><body>
  <h2>Mi colección de cds</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Titulo</th>
      <th>Artista</th>
    </tr>
    <tr>
      <td>Empire Burlesque</td>
      <td>Bob Dylan</td>
    </tr>
    …
    <tr>
      <td>1999 Grammy Nominees</td>
      <td>Many</td>
    </tr>
  </table>
</body></html>
```

XSLT

- Resultado

Mi colección de cds

Titulo	Artista
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many

XSLT

- Soporte de browsers
  - La mayoría de los exploradores soportan XML y XSLT
  - Mozilla Firefox
    - Soporta XML y XSLT dede la versión 3
  - Internet Explorer
    - Soporta XML, XSLT y XPath desde la versión 6
  - Google Chrome
    - Soporta XML, XSLT y XPath desde la versión 1
  - Opera
    - Soporta XML, XSLT y XPath desde la versión 9
  - Safari
    - Soporta XML y XSLT desde la versión 1

Sintaxis XSLT

- El elemento raíz en un XSLT debe ser uno de
  - <xsl:stylesheet>
  - <xsl:transform>
- Son sinónimos y cualquiera se puede usar
- Ejemplos validos:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
...
</xsl:stylesheet>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
...
</xsl:transform>
```

Sintaxis XSLT

- Para tener acceso a los elementos, atributos y funciones XSLT debemos declarar el namespace al comienzo

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

- Como se referencia un archivo xslt desde un xml para ser transformado en un browser?

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

Sintaxis XSLT

- Un archivo XSLT consiste de una o mas reglas de transformación, las cuales se denominan templates
- El elemento <xsl:template> define un template
  - Define un atributo match el cual es usado para identificar una sección dentro del documento XML de entrada en la transformación
  - Para utilizar la totalidad del documento de entrada para la transformación se define el carácter reservado “/”
  - El contenido de este elemento es lo que se va a dar como resultado de la transformación, es decir, será el XML output
- Ejemplo

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

Sintaxis XSLT

- El elemento <xsl:value-of> es utilizado para extraer el valor de un elemento del XML input, y adicionarlo al XML output
- Define un atributo select
  - Valor es una expresión Xpath
- Ejemplo

```
<xsl:value-of select="cd/titulo"/>
```

Sintaxis XSLT

- Elementos de control
  - <xsl:for-each>
  - <xsl:sort>
  - <xsl:if>
  - <xsl:choose>, <xsl:when> y <xsl:otherwise>

Sintaxis XSLT

- El elemento <xsl:for-each> permite loopear
- Define el atributo select
  - Expresión XPath para seleccionar el conjunto de nodos sobre los cuales loopear

- Ejemplo

```
<xsl:for-each select="catalog/cd[artist='Bob Dylan']">
  <tr>
    <td><xsl:value-of select="title" /></td>
    <td><xsl:value-of select="artist" /></td>
  </tr>
</xsl:for-each>
```

Sintaxis XSLT

- El elemento <xsl:sort> esta diseñado para ordenar el XML output
- Se utiliza dentro de un loop <xsl:for-each>
- Define un atributo select que debe indicar un elemento XML que oficiara como criterio de ordenación
- Ejemplo:

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

Sintaxis XSLT

- El elemento <xsl:if> se utiliza para establecer condiciones booleanas sobre un determinado elemento o atributo del XML input
- Define un atributo test cuyo valor es una expresión XPath cuyo resultado debe ser un valor booleano
- Ejemplo:

```
<xsl:for-each select="catalog/cd">
  <xsl:if test="price &gt; 10">
    ...
  </xsl:if>
</xsl:for-each>
```

Sintaxis XSLT

- Los elementos <xsl:choose>, <xsl:when> y <xsl:otherwise> son utilizados para expresar múltiples evaluaciones condicionales
- Ejemplo:

```
<xsl:for-each select="catalog/cd">
  <tr>
    <td><xsl:value-of select="title" /></td>
    <xsl:choose>
      <xsl:when test="price &gt; 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>
```

Sintaxis XSLT

- Pueden haber tantos <xsl:when> como se desee
  - Las expresiones no tienen por que se mutuoexcluyentes
  - Se evalúan en orden de ocurrencia todas, pero se incluyen los condicionales anteriores
- Ejemplo

```
<xsl:choose>
  <xsl:when test="price &gt; 10">
    <td bgcolor="#ff00ff"><xsl:value-of select="artist" /></td>
  </xsl:when>
  <xsl:when test="price &gt; 9">
    <td bgcolor="#cccccc"><xsl:value-of select="artist" /></td>
  </xsl:when>
  <xsl:otherwise>
    <td><xsl:value-of select="artist" /></td>
  </xsl:otherwise>
</xsl:choose>
```

- La fila que contiene el valor del artista, se pintara de rosado cuando el precio del cd sea mayor a 10, y de color gris cuando el precio del cd sea mayor a 9 y menor a 10, de lo contario no se le pone fondo

Sintaxis XSLT

- El elemento <xsl:apply-templates> es utilizado para aplicar un template a un elemento o a los hijos de dicho elemento
- Define opcionalmente un atributo select, expresión XPath, que se utiliza para filtrar sobre cuales elementos aplicar el template

Sintaxis XSLT

- Ejemplo de <xsl:apply-templates>

```
<?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>
      <xsl:apply-templates />
    </body>
  </html>
</xsl:template>

<xsl:template match="cd">
  <p>
    <xsl:apply-templates select="title" />
    <xsl:apply-templates select="artist" />
  </p>
</xsl:template>
```

```
<xsl:template match="title">
  Title:
  <span style="color:#ff0000">
    <xsl:value-of select="." />
  </span>
  <br />
</xsl:template>

<xsl:template match="artist">
  Artist:
  <span style="color:#00ff00">
    <xsl:value-of select="." />
  </span>
  <br />
</xsl:template>

</xsl:stylesheet>
```

Otros elementos XSLT

- <xsl:attribute>
  - Agrega un atributo a un elemento

```
- <picture>
    <xsl:attribute name="source"/>
  </picture>

- <picture>
    <xsl:attribute name="source">
      <xsl:value-of select="images/name"/>
    </xsl:attribute>
  </picture>
```

Otros elementos XSLT

- <xsl:copy>
  - Copia el elemento actual sin hijos
- <xsl:copy-of>
  - Copia el elemento actual con hijos
- <xsl:fallback>
  - Especifica un codigo alternativo para utilizar si el procesador XSLT no soporta un determinado elemento XSLT
  - Se pone como hijo del elemento xsl que se piensa no soportado

Otros elementos XSLT

- <xsl:message>
  - Escribe un mensaje el output stream
  - Utilizado para reportar errores

```
<xsl:if test="artist=''">
  <xsl:message terminate="yes">
    Error: Artist is an empty string!
  </xsl:message>
</xsl:if>
```

Otros elementos XSLT

- <xsl:output>
  - Define el formato del XML output
  - Debe aparecer como hijo del elemento <xsl:stylesheet> o <xsl:transform>
  - Sintaxis:

```
<xsl:output method="xml|html|text|name"
  version="string"
  encoding="string"
  omit-xml-declaration="yes|no"
  standalone="yes|no"
  doctype-public="string"
  doctype-system="string"
  cdata-section-elements="namelist"
  indent="yes|no"
  media-type="string" />
```

Otros elementos XSLT

- <xsl:variable>
  - Declara una variable local o global
    - Si es global debe ser hijo del elemento <xsl:stylesheet> o <xsl:transform>
    - Si es local debe ser hijo de <xsl:template>
  - Una vez seteado el valor de una variable no puede ser modificado
    - Son constantes
  - Ejemplo:
    - `<xsl:variable name="color" select="'red'" />`
    - `<xsl:variable name="color" select='"red"' />`

Otros elementos XSLT

- <xsl:variable>
  - Ejemplo:

```
<xsl:variable name="header">
  <tr>
    <th>Element</th>
    <th>Description</th>
  </tr>
</xsl:variable>
…
<xsl:copy-of select="$header" />
```

Editores XSLT on-line

- http://www.w3schools.com/xsl/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog