1 ASIX

**APUNTES EXAMEN XML // XSD // XSL**

línea horizontal

# 



# XSD

Declaramos xml y xsd:

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

xs:element → elemento del documento

xs:complexType → dentro de un elemento hay varios elementos

xs:sequence → lo mismo que el complex

* debajo de la etiqueta final de sequence podemos poner esto:
  + xs:attribute → atributos de los elementos

xs:simpleType → sirve para poner restricciones a los elementos

dentro del simpleType tenemos:

* xs:restriction → declara el inicio de una restricción
* xs:ennumeration → declara que la restricción puede tener los valores que tenga el ennumeration
* xs:maxLength → longitud máxima de una cadena de caracteres
* xs:minLength → longitud mínima de una cadena de caracteres

# XSL

Para empezar el documento declaramos que es un documento xml y xsl.

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

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

declaramos el xsl:template:

xsl:template match=”/”

(contiene las reglas necesarias para que funcione el xsl)

declaramos después el html, y dentro del html el head y dentro del head el style, tal que así.

* html
  + head
    - style
  + body

dentro del body podemos poner cosas del html como por ejemplo el h1, p, a, img, etcétera.

también tenemos el foreach

* selecciona datos de ciertas etiquetas, por ejemplo:
  + <xsl:for-each select=”dades/dada”>

también tenemos el choose

* xs:choose → permite usar el xs:when

el xs:when es como el if de c++, es para que se cumplan ciertas condiciones.

* <xsl:when test=”tipus = ‘Serie’>

el class es un atributo que se coloca a los elementos del xsl para poder seleccionar en el css y otros por el contenido que hay dentro de la etiqueta esa en concreto:

* <tr class = ‘{genere}’></tr>