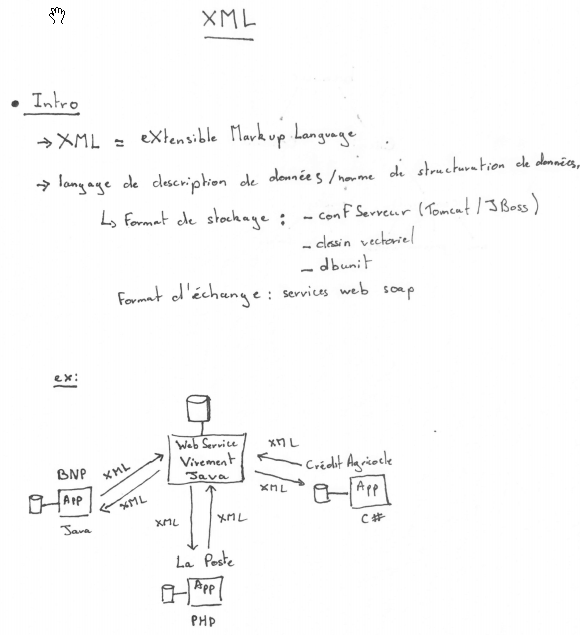
**Cours du 2 janvier**

**. XML .**

Professeur : T. de Mijola



eXtensible Markup Language

Utillité :  
  
Décrire et stocker les données :  
- configuration : de serveur (Tomcat, JBoss), de projet (pom.xml, pour Maven)  
- base de données

Transfert des données (objets) via le réseau (fichier wsdl)

Formalisme, donc lisible sur tout ordinateur.

Le pom.xml sert à Maven (compilateur du projet et import des dépendances), sert au projet J2EE

dbUni, utilisé par SquashTM, pour tester une base de données (automatisation de tests) 🡪 XML permet donc de décrire une base de données.

XML étant un standard (normes & spécifications actées par un consortium, le W3C), il est « parlé » par tous les autres langages : cela permet de transférer l’information entre les différents langages.

Langage JSON

Ligne en-tête d’un XML : < ?xml version= ‘’1.0’’ encoding=’’UTF-8’’ ?>

SGML

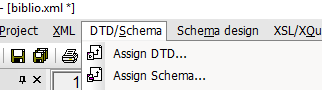
XML HTML  
DTD XHTML  
XSD  
XSL

Le DOCTYPE (ligne d’en tête) est en SGML.  
  
Ensuite, une balise racine exemple :  
<project attribut1=’’ attribut1’’, attribut2=’’ attribut2’’> </project>

xmlns 🡪 xml namespace, pour indiquer les conventions utilisées et pour vérifier leur validité.  
Les commentaires sont comme en html < !-- commentaire -->

Les balises sont sensibles à la casse.  
Un nom xml doit commencer par un underscore, ou un caractère, et ne peut commencer par les lettres xml. Par ailleurs, ne peut non plus contenir d’entités réservées ‘’ < > & ‘

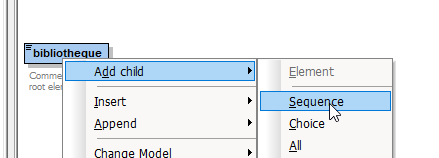
API DOM : Document Object Model ([clic](https://fr.wikipedia.org/wiki/Document_Object_Model))  
Lisibilité, en dessinant une structure avec des balises wrapper.

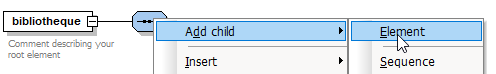


Génère automatiquement cette ligne :

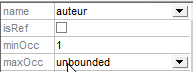


Schema :

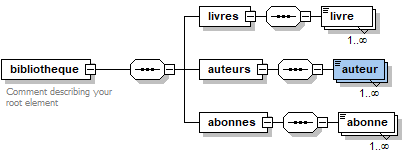




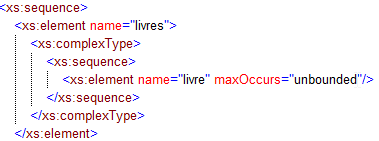
On ajoute les cardinalités aux enfants :



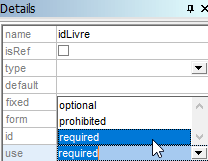
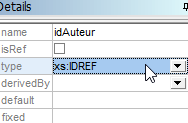
Le schéma s’adapte :



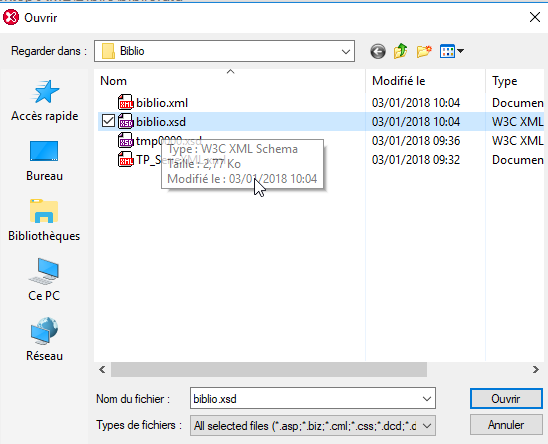
Ainsi que le texte :



Ajout de l’attribut, et pose en REQUIRED :

 Et type IDREF : 

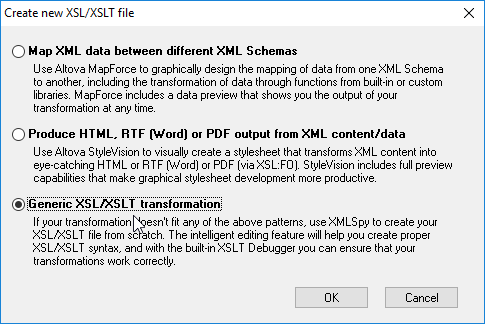
Après avoir tout ajouté, on fait DTD /SCHEMA et on add :



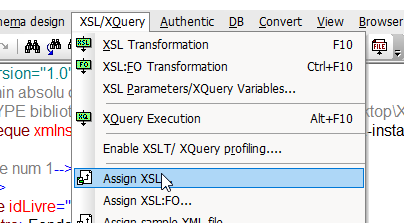
Création d’un xsl :

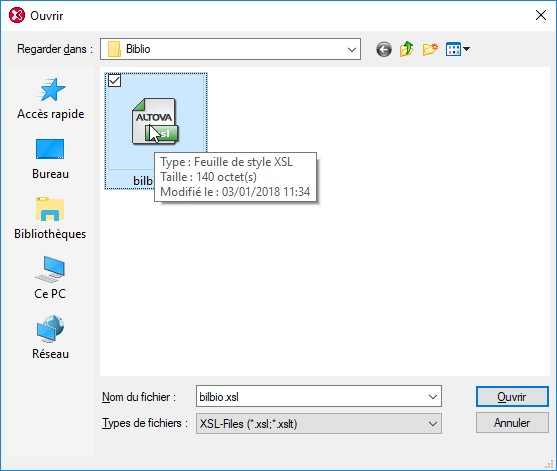


Option numéro 3 :

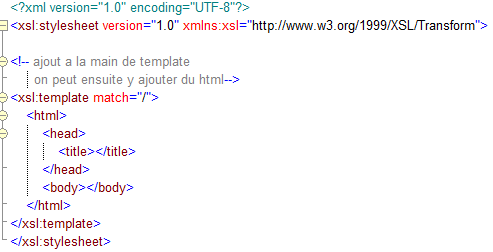


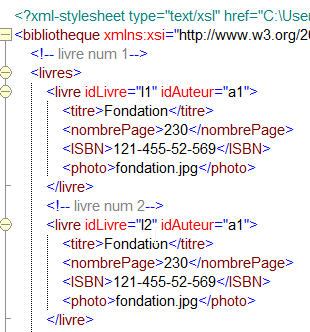
On lie notre xml au xsl tout fraichement créé :



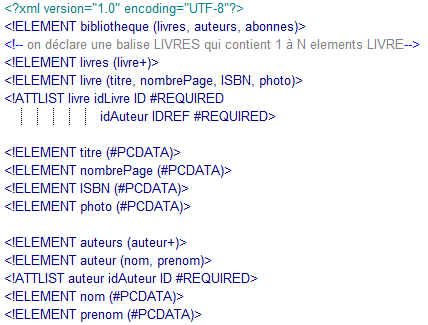


On travaille sur la feuille XSL :



Reprise des captures d’écran :  
  
XLM, l’exemple du livre :  


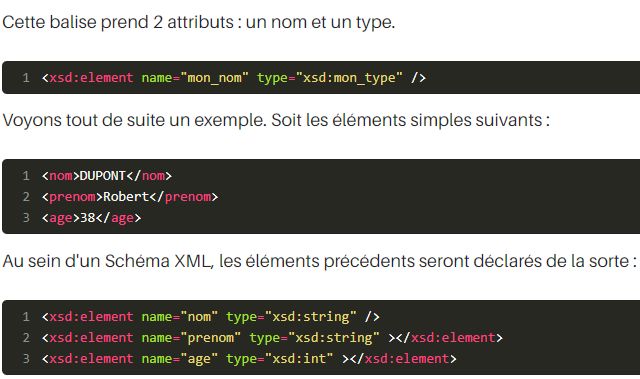
Il devient en dtd :



**. XSD .**

23 mars 2018 :

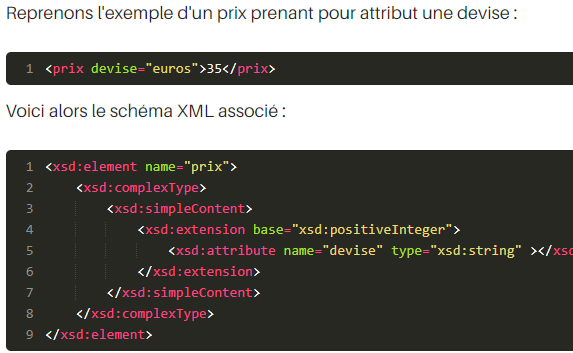
Balises simples :

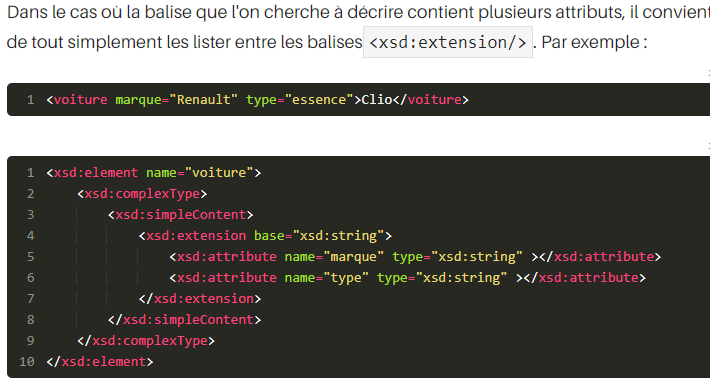


Balise par défaut : default := « val »  


Valeur fixée :  
 

Attributs et définition xsd :



Extension donner les attributs.  


<xsd :all> et les attributs dans cette balise : ils doivent tous apparaître, mais pas d’ordre imposé.  
<xsd :sequence> : les attributs doivent apparaitre dans cet ordre.  
<xsd :choice> les attributs ne peuvent tous apparaitre : un seul.