

Bonjour tout le monde



Password

<https://1drv.ms/u/s!AvHBznCKgzcSoLZU1kNLh2cp4YQMZQ?e=jhmKRj>

Contenu du cours : <https://www.ib-formation.fr/catalogue/nbs-details/catref/universib-formations-informatiques-developpement-web-et-mobilite-soa-woa-esb-xml-api-rest/ref/ob305/xml-au-coeur-du-systeme-dinformation>

① XML ? → ② bien formé ?

→ valide ?

→ Éditeurs →

→ schéma(s) - namespaces

Applications du XML

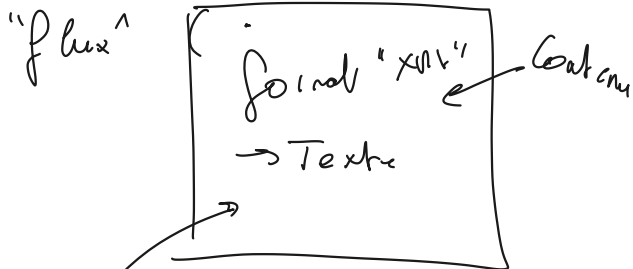
→ XSLT (Technique!)
Transformation des données spécifiques

→ SVG (graphiques vectoriels)

→ XPath / XQuery

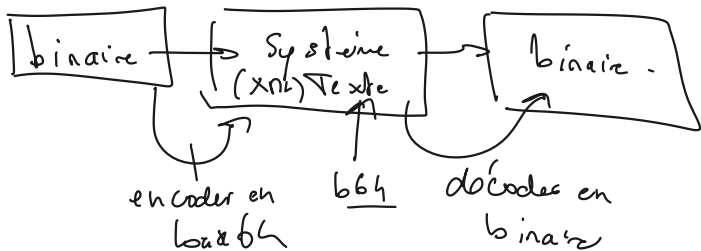
• Outils Altova

• Différences par rapport aux BDD



Contenu binaire (image, vidéo, etc...)

→ Encodé en "base 64"



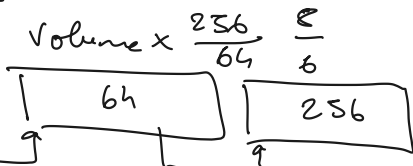
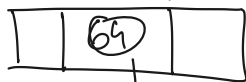
1 octet / valeur

que de l'imprimable

256 possibilités
1 octet

→ imprimables (texte)

→ non imprimables
imprimable



256 / octet.

Tables 1 octet / charac → ASCII

2-4 octets / char → Unicode → "tranquille"

ASCII

180 8859 - 2

Table 1 -- 127 → Common

- france
 - allemagne
 - grèce
 - etc...
- } caractères spéciaux.

128 .. 255 → en fonction de la culture.

180 8859 - 7

XNL → Stocker des Contenu

Contenu Texte ↙
↳ encapsuler dans
des XML

↘ Données
Excellents
Pour
le
Contenu
Complexe,
Structure

Compatible
Excel

"Tabulaires"

— ~~pas~~ Pas de XML

— zone

variable: valeur: Contenu simple
(ex: 1 cellule
soit Excel)

X

Y

→ Tableau -

Contenu:

clé1: valeur

clé2: valeur

clé3:

clé1: valeur

clé2: valeur 2

etc...

→ Complexe,
Structure

ex

exemple de données complexes:

Commande d'un client -

→ données du client (Nom, prénom,
adresse)

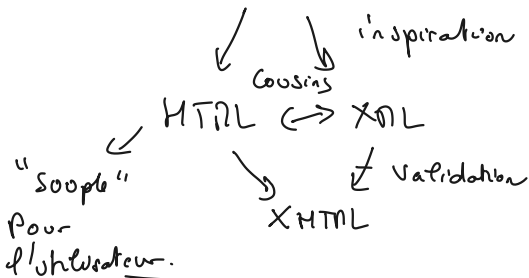
→ " de la commande (date, numéro,
frais de port, ...)

→ liste de produits de la commande (légende: Nom Prod,
Prix, Qté)

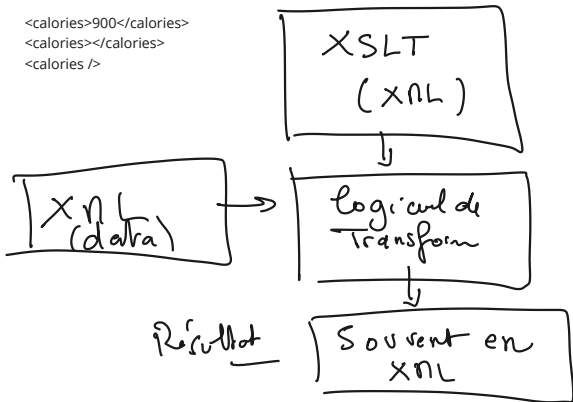
XNL → Technique, strict
→ sensible à la casse

≈ 70

SGML



<calories>900</calories>
<calories></calories>
<calories />



XNL "bien formé"

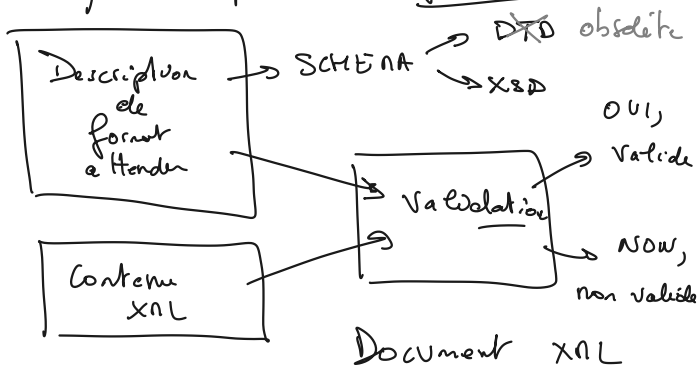
→ est-ce qu'il y a des erreurs de syntaxe (pas de contenu)

→ intégrer

↓
NON → réglé

↓
OUI : est-ce que l'on entretient
compte -

On peut s'attendre à un contenu qui respecte un format attendu

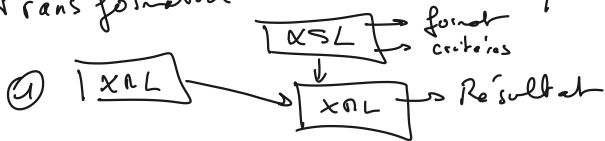


Document XNL

Valide,

qui respecte le Schema

Transformation de XML style SQL



② Utilisation de logiciels d'édition "boîte à outils"

ex. XML Spy → Apprendre le logiciel

③ Utilisation / Développement de logiciels / exploitation du DOM (Doc. Object. Model)

<https://mkyong.com/python/python-read-xml-file-dom-example/>

avec JavaScript → ou Python / DOM

④ Utilisation d'outils ETL

ex. Talend → Éditeur graphique
Process
XML / XSL

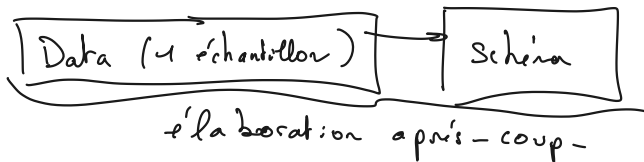
- 1) Installez Visual Studio Code <https://code.visualstudio.com/Download>
 - 2) Télécharger le fichier northwind.xml
 - 3) ouvrez-le avec vscode
-

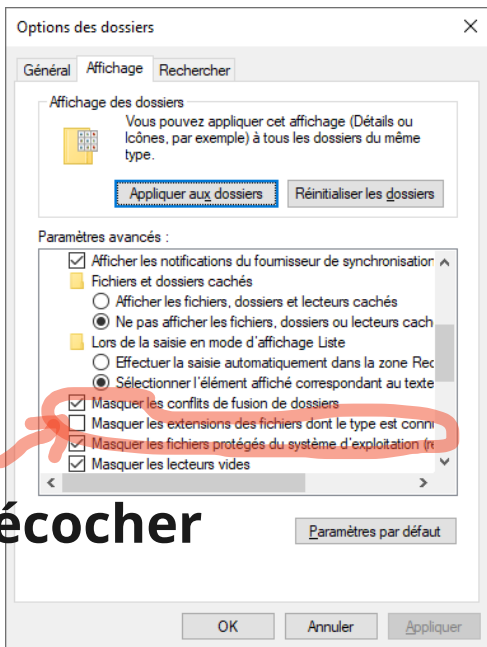
Déclarer un Process pour les flux XML



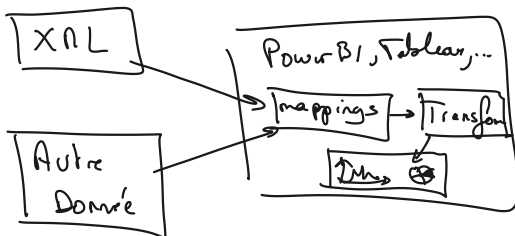
↓
Valide (schéma)

↳ Specs en amont → ex: liste de pays





Décocher



ISO → Ascii
UTF → Unicode

ligne recommandée

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!-- Date de création : 30/09/07 -->  
<cours titre="XML">  
  <intervenant nom="alexandre brillant">  
  </intervenant>  
  <plan>  
    Introduction  
    XML et la composition de documents  
  </plan>  
</cours>
```

Comments

Contenu

attribut
claf = "valeur"

Comment <!-- -->

! Pas de sous-commentaires!

l'élément Racine du document

! Il doit être unique

- intervenant : Sous-éléments

- plan - Contenu
- Attributs

- Contenus <balise> ... </balise>
Contenu

Types de données

- Simples
- Tabulaires (champs & plot)
- Complexes (structure, sous structure, etc)

Balise, Tag, Element

↳ données simples, (Tab),
ou complexes

a Attribut

↳ données de Type string

String → chaîne
entier
flottant $1.e+15$
dates

Data → stocker 1 information

Complexe → Dans 1 élément

Simple → Attribut → "léger"
1 élément → "structure"

A attribut = "sympa" pour de me la données -

```
<?xml version="1.1" encoding="UTF-8"?>
<commandes>
  <commande idcde="1" idcli="1">
  </commande>
  <!-- ou :
    <commande idcde="1">
      <client id="1" />
    </commande>
  ou :
    <client><id>1</id></client>
  </commande>
  <!-->
</commandes>
```

```
<?xml version="1.1" encoding="UTF-8"?>
<clients>
  <client idcli="1">
    <nom>A</nom>
  </client>
  <client idcli="2">
    <nom>B</nom>
  </client>
</clients>
```

Exercice 1

Création d'un livre en XML

On souhaite écrire un livre en utilisant le formalisme XML. Le livre est structuré en sections (au moins 2), en chapitres (au moins 2) et en paragraphes (au moins 2).

Le livre doit contenir la liste des auteurs (avec nom et prénom).

Tous les éléments doivent posséder un titre, sauf le paragraphe qui contient du texte.

Proposez une structuration XML de ce document (avec 2 auteurs, 2 sections, 2 chapitres par section et 2 paragraphes par chapitre).

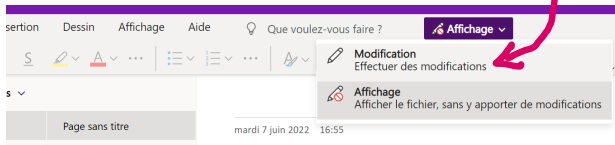
Vérifiez, à l'aide de l'éditeur, que votre document est bien formé.

Votre document sera nommé `livre1.xml`.

Lien vers le bloc notes partagé d'exercices/solutions :

https://1drv.ms/u/s!AvHBznCKgzCSoLZ_-1Qs7ehRmZaT1Q?e=YTghRB

Passer en modification



lower casing

UPPERCASING

Pascal Casing

camel Casing

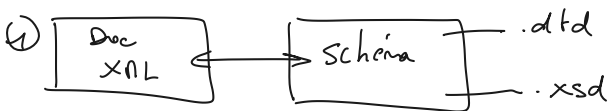


```
<?xml version="1.0" encoding="UTF-8"?>
<Bibliotheque>
  <Auteurs>
    <Auteur AuteurID="1">
      <LastName>Davolio</LastName>
      <FirstName>Nancy</FirstName>
      <TitleOfCourtesy>Ms.</TitleOfCourtesy>
      <BirthDate>1948-12-08</BirthDate>
      <HireDate>1992-05-01</HireDate>
    </Auteur>
    <Auteur AuteurID="2">
      <LastName>Toto</LastName>
      <FirstName>Beta</FirstName>
      <TitleOfCourtesy>Ms.</TitleOfCourtesy>
      <BirthDate>1952-11-18</BirthDate>
      <HireDate>1972-05-11</HireDate>
    </Auteur>
  </Auteurs>
  <Livres>
    <Livre AuteurID="1">
      <Chapitres>
        <Chapitre ChapitreID="1">
          <ChapterName>TITRE</ChapterName>
          <ChapterNameName>SOUS-TITRE</ChapterNameName>
        </Chapitre>
        <Chapitre ChapitreID="2">
          <ChapterName>TITRE</ChapterName>
          <ChapterNameName>SOUS-TITRE</ChapterNameName>
        </Chapitre>
      </Chapitres>
      <Paragraphes>
        <Paragraphe ParagrapheID="1">
          <ParagrapheTexte>ehhiodhfurgshlkduhguizkghsoidugh</ParagrapheTexte>
        </Paragraphe>
        <Paragraphe ParagrapheID="2">
          <ParagrapheTexte>ljsioeifoidghsdli</ParagrapheTexte>
        </Paragraphe>
      </Paragraphes>
    </Livre>
  </Livres>
</Bibliotheque>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<Bibliotheque>
  <Auteurs>
    <Auteur AuteurID="1">
      <LastName>Davolio</LastName>
      <FirstName>Nancy</FirstName>
      <TitleOfCourtesy>Ms.</TitleOfCourtesy>
      <BirthDate>1948-12-08</BirthDate>
      <HireDate>1992-05-01</HireDate>
    </Auteur>
    <Auteur AuteurID="2">
      <LastName>Toto</LastName>
      <FirstName>Beta</FirstName>
      <TitleOfCourtesy>Ms.</TitleOfCourtesy>
      <BirthDate>1952-11-18</BirthDate>
      <HireDate>1972-05-11</HireDate>
    </Auteur>
  </Auteurs>
  <Livres>
    <Livre AuteurID="1">
      <Chapitres>
        <Chapitre ChapiteID="1">
          <ChapterName>TITRE</ChapterName>
          <ChapterNameName>SOUS-TITRE</ChapterNameName>
        </Chapitre>
        <Chapitre ChapiteID="2">
          <ChapterName>TITRE</ChapterName>
          <ChapterNameName>SOUS-TITRE</ChapterNameName>
        </Chapitre>
      </Chapitres>
      <Paragraphes>
        <Paragraphe ParagrapheID="1">
          <ParagrapheTexte>ehhiodhfeurgshlkduhguizkgshsoidugh</ParagrapheTexte>
        </Paragraphe>
        <Paragraphe ParagrapheID="2">
          <ParagrapheTexte>ljsoeifoidghsdli</ParagrapheTexte>
        </Paragraphe>
      </Paragraphes>
    </Livre>
  </Livres>
</Bibliotheque>

```



<https://www.w3.org/XML/1998/06/xmlspec-report-19980910.htm>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Bibliotheque SYSTEM "livres.dtd">
<Bibliotheque>
<Auteurs>
```

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

<!-- DTD generated by XMLSpy v2022 rel. 2 (x64) (http://www.altova.com) -->
<ELEMENT Livre (Chapitres, Paragraphes)>
<!ATTLIST Livre
    AuteurID CDATA #FIXED "1"
>
<ELEMENT Auteur (LastName, FirstName, TitleOfCourtesy, BirthDate, HireDate)>
<!ATTLIST Auteur
    AuteurID CDATA #REQUIRED
>
<ELEMENT Livres (Livre)>
<ELEMENT Auteurs (Auteur+)>
<ELEMENT Chapitre (ChapterName, ChapterNameName)>
<!ATTLIST Chapitre
    ChapiteID (1 | 2) #REQUIRED
>
<ELEMENT HireDate (#PCDATA)>
<ELEMENT LastName (#PCDATA)>
<ELEMENT BirthDate (#PCDATA)>
<ELEMENT Chapitres (Chapitre+)>
<ELEMENT FirstName (#PCDATA)>
<ELEMENT Paragraphe (ParagrapheTexte)>
<!ATTLIST Paragraphe
    ParagrapheID (1 | 2) #REQUIRED
>
<ELEMENT ChapterName (#PCDATA)>
<ELEMENT Paragraphes (Paragraphe+)>
<ELEMENT Bibliotheque (Auteurs, Livres)>
<ELEMENT ChapterNameName (#PCDATA)>
<ELEMENT ParagrapheTexte (#PCDATA)>
<ELEMENT TitleOfCourtesy (#PCDATA)>
```


← → ↻ w3.org/2001/XMLSchema

flux du schéma,
inexploitable

XML Schema

15 October 2014

Table of contents

1. [Introduction](#)
2. [Resources](#)

Introduction

This document describes the [XML Schema](#) namespace. It also contains a directory of links to these

Related Resources for XML Schema

Schemas for XML Schema

```
<!-- W3C Schema generated by XMLSpy v2022 rel. 2 (x64) (http://www.altova.com) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Auteur">
    <xs:complexType>
      <xs:sequence>
```

Permet de reconnaître que

c'est un schéma!
"marqueur, un identifiant"

```
<Bibliothèque xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="bibliotheque.xsd">
  <Auteurs>
    <Auteur AuteurID="1">
      <LastName>Davalio</LastName>
      <FirstName>Nancy</FirstName>
      <TitleOfCourtesy>Ms.</TitleOfCourtesy>
      <BirthDate>1948-12-08</BirthDate>
      <HireDate>1992-05-01</HireDate>
    </Auteur>
    <Auteur AuteurID="4">
      <LastName>Toto</LastName>
```

déclaration de schéma sans namespace
équivalent de:
xmlns="bibliotheque.xsd"

soit notion de namespace 2 pages + loin →

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- provide ID type information even for parsers which only read the
internal subset -->
<!--
    Make sure that processors that do not read the external
    subset will know about the various IDs we declare
-->
<?xml:
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" xml:lang="EN" targetNamespace="http://www.w3.org/2001/XMLSchema"
  <xs:annotation>
    <xs:documentation> Part 1 version: structures.xsd (rec-20120405) Part 2 version: datatypes.xsd (rec-20120405) </xs:documentation>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation source="../structures/structures.html"> The schema corresponding to this document is normative, with respect
    below, is not normative, but rather highlights important aspects of the W3C Recommendation of which this is a part. See below
  </xs:documentation>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation> The simpleType element and all of its members are defined towards the end of this schema document.</xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://www.w3.org/2001/xml.xsd"
  <xs:annotation>
    <xs:documentation> Get access to the xml: attribute groups for xml:lang as declared on 'schema' and 'documentation' below </
    </xs:documentation>
  </xs:import>
  <xs:complexType name="openAttrs">
    <xs:annotation>
      <xs:documentation> This type is extended by almost all schema types to allow attributes from other namespaces to be added to
    </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="annotated">
    <xs:annotation>
      <xs:documentation> This type is extended by all types which allow annotation other than <schema> itself </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:sequence>
          <xs:element ref="xs:annotation" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:group name="composition">
    <xs:choice>
      <xs:element ref="xs:include"/>
      <xs:element ref="xs:import"/>
      <xs:element ref="xs:redefine"/>
      <xs:element ref="xs:override"/>
      <xs:element ref="xs:annotation"/>
    </xs:choice>
  </xs:group>
  <xs:group name="schemaTop">
    <xs:annotation>
```

Namespaces

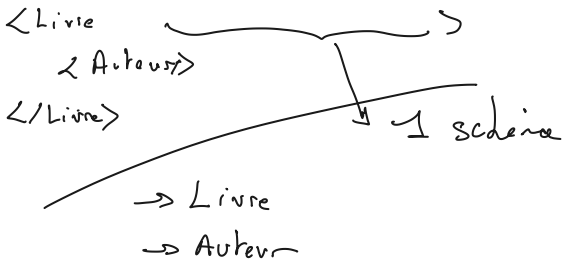
Ex dossier racine - dossier racine (pas de nom)
- sous dossiers (1 nom)
(1 niveau) (pas de sous-sous dossiers)

les fichiers (contenu) du dossier racine ou
sous dossiers ne se voient pas -
* il peut y avoir un flux par
dossiers (racine ou sous dossier)

Equivalent des dossiers et flux XML
sont les espaces de nommage (namespaces)
qui portent un nom (racine ou
spécifique) ³ Pas de nom.
↳ nommé
Ils ont un schéma qui sera attribué
aux éléments ou attributs

Exemple 1

flux simple



clients.xml

```
<client id="..."
  <Adresse
  <commentaire
```

clients.xml

```
- client
- Adresse
- commentaire
```

Commandes.xml

```
<Commandes>
  <Commande>
    <Produit>
```

Commandes.xml

```
Commandes
Commande
```

Produits de la

```
Commande.xml
<Produit
  Qte/Code
```

Produits.xml

```
<Produit>
  <Nom
  <Prix>
```

Produit.xml
PrixCatalog

namespace "racine" sans nom

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- <Bibliothèque xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="bibliotheque.xsd" -->
3 <!-- ici tout le contenu est défini dans le namespace racine ( les éléments et attributs ne seront pas préfixés par un namespace )
4 <Bibliothèque xmlns="bibliotheque.xsd">
5   <Auteurs>
6     <Auteur AuteurID="1">
7       <LastName>Davolio</LastName>
8       <FirstName>Nancy</FirstName>
9     </Auteur>
10  <!-- ici tout le contenu est défini dans le namespace truc ( les éléments et attributs seront préfixés par le ns truc ) -->
11  <truc:Bibliothèque xmlns:truc="bibliotheque.xsd">
12    <truc:Auteurs>
13      <truc:Auteur AuteurID="1">
14        <truc:LastName>Davolio</truc:LastName>
15        <truc:FirstName>Nancy</truc:FirstName>
16      </truc:Auteur>
17    </truc:Auteurs>
18  </truc:Bibliothèque>
19 </Bibliothèque>

```

export de ma bdd. xml

< Export >

< Clients >

< client xmlns="client.xsd" >

< Produits >

prod: Product xmlns:prod="product.xsd"

prod: Price prod: QtyInStock prod: QtyOrdered

< Commandes >

< commande xmlns="commande.xsd" >

prod: Product xmlns:prod="prodde la commande.xsd"

prod: Price prod: Qty prod: QtyOrdered

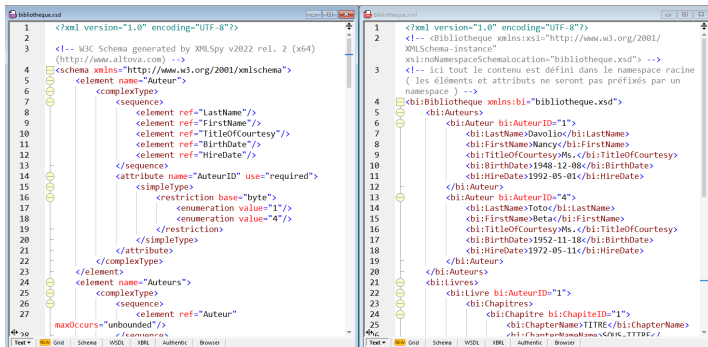
< Product

Pas la même structure!

Lien vers le bloc notes partagé d'exercices/solutions :

https://1drv.ms/u/s!AvHBznCKgzcSolZ_-1Qs7ehRmZaT1Q?e=YTghRB

Exercice Schémas : définissez un namespace pour la bibliothèque, et simplifiez le namespace dans le xsd



Exemple d'utilisation de plusieurs namespaces dont schéma par défaut avec WPF :

```
1 <Window x:Class="CalculatorDemo.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:CalculatorDemo"
7     mc:Ignorable="d"
8     Title="MainWindow"
9     Height="350"
10    Width="525"
11    Icon="appicon.ico">
12    <DockPanel Name="MyPanel">
13        <Menu DockPanel.Dock="Top" Height="26">
14            <MenuItem Header="File">
15                <MenuItem Click="OnMenuExit" Header="Exit" />
16            </MenuItem>
17            <MenuItem Header="View">
18                <MenuItem Name="StandardMenu" Click="OnMenuStandard" IsCheckable="true" IsChecked="True"
19                    Header="Standard" />
20            </MenuItem>
21            <MenuItem Header="Help">
22                <MenuItem Click="OnMenuAbout" Header="About" />
23            </MenuItem>
24        </Menu>
25        <Grid Name="MyGrid" ShowGridLines="False">
26            <Grid.ColumnDefinitions>
```

Solution de l'exercice :

```
<?xml version="1.1" encoding="UTF-8"?>
<!-- <backup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSc
<backup
  xmlns="produits.xsd"
  xmlns:ptdcde="produits2.xsd"
>
  <produits>
    <produit id="1">
      <nom comment="le premier produit">Chai</nom>
      <prix>10</prix>
    </produit>
    <produit id="2">
      <nom>Chang</nom>
      <prix>12</prix>
    </produit>
  </produits>
  <ptdcde:commandes>
    <ptdcde:commande>
      <ptdcde:produit>
        <ptdcde:id>1</ptdcde:id>
        <ptdcde:name>Chai</ptdcde:name>
        <ptdcde:prix>10</ptdcde:prix>
        <ptdcde:qtecommandee>3</ptdcde:qtecommandee>
      </ptdcde:produit>
    </ptdcde:commande>
  </ptdcde:commandes>
</backup>
```

Expressions régulières :

[a-z] : un caractère, a à z (une lettre en minuscule)

[0-9] : un chiffre, n'importe lequel

[1-4] : un chiffre, 1 ou 2 ou 3 ou 4

+ : 1 à n occurrences

* : 0 à n occurrences

[acdef] : a ou c ou d ou e ou f

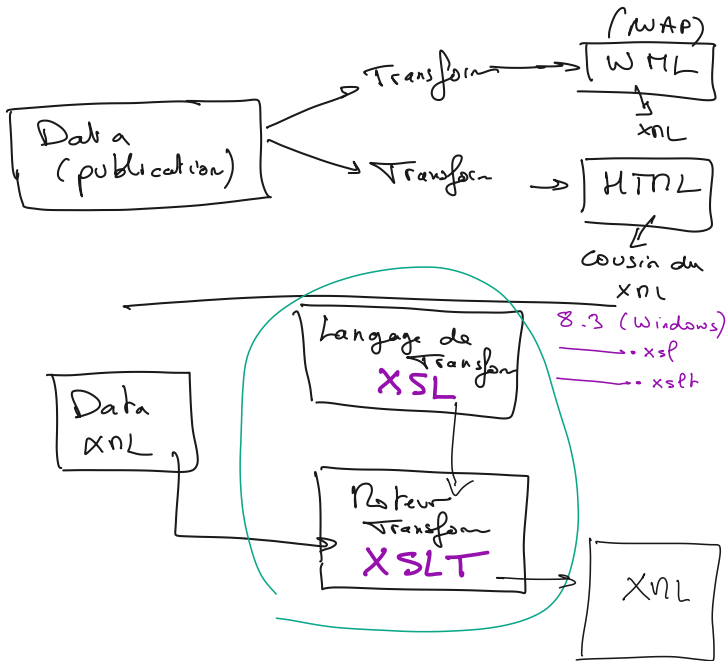
[a-zA-Z] : toute lettre

[0-9a-zA-Z] : toute caractère alphanumérique

a vérifier : [0-9^5] : tout chiffre sauf 5

[]^5 : 2 caractères n'importe lesquels mais le second ne peut pas être un 5

Référence vers les XSD : https://www.w3schools.com/xml/schema_elements_ref.asp



Exemple de transformation XML vers HTML :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <catalog>
3   <cd>
4     <title>Empire Burlesque</title>
5     <artist>Bob Dylan</artist>
6     <country>USA</country>
7     <company>Columbia</company>
8     <price>10.90</price>
9     <year>1985</year>
10  </cd>
11  <cd>
12    <title>Empire Burlesque 2</title>
13    <artist>Bob Dylan</artist>
14    <country>USA</country>
15    <company>Columbia</company>
16    <price>10.90</price>
17    <year>1985</year>
18  </cd>
19 </catalog>
```

XSL :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <xsl:stylesheet version="1.0"
4   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
5   <xsl:template match="/">
6     <html>
7       <body>
8         <h2>My CD Collection</h2>
9         <table border="1">
10           <tr bgcolor="#9acd32">
11             <th>Title</th>
12             <th>Artist</th>
13           </tr>
14           <xsl:for-each select="catalog/cd">
15             <tr>
16               <td><xsl:value-of select="title"></td>
17               <td><xsl:value-of select="artist"></td>
18             </tr>
19           </xsl:for-each>
20         </table>
21       </body>
22     </html>
23   </xsl:template>
24 </xsl:stylesheet>
```

env toi
quel

envoyer' toi
quel

Résultat :

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Empire Burlesque 2	Bob Dylan
Empire Burlesque 3	Bob Dylan
Empire Burlesque 4	Bob Dylan
Empire Burlesque	Bob Dylan
Empire Burlesque	Bob Dylan

```
1 <?xml?>
2 <body>
3   <h2>My CD Collection</h2>
4   <table border="1">
5     <tr bgcolor="#9acd32">
6       <th>Title</th>
7       <th>Artist</th>
8     </tr>
9     <tr>
10      <td>Empire Burlesque</td>
11      <td>Bob Dylan</td>
12    </tr>
13  </table>
14 </body>
15 </html>
```

Génère :

```
INSERT INTO ARTISTS (title,artist) VALUES ('Empire Burlesque','Bob Dylan');
INSERT INTO ARTISTS (title,artist) VALUES ('Empire Burlesque 2','Bob Dylan');
```

```
1 <?xml:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"?>
2 <xsl:output omit-xml-declaration="yes" indent="yes"/>
3 <xsl:template match="/"><xsl:for-each select="catalog/cd">INSERT INTO ARTISTS (title,artist) VALUES ('<xsl:value-of select="title"/>','<xsl:value-of
4   select="artist"/>');<xsl:text>&#10;</xsl:text></xsl:for-each></xsl:template></xsl:stylesheet>
```

Génère :

title;artist

Empire Burlesque;Bob Dylan

Empire Burlesque 2;Bob Dylan

Référence du contenu SVG :

<https://developer.mozilla.org/fr/docs/Web/SVG/Element>

Editeur SVG en ligne :

https://www.w3schools.com/graphics/tryit.asp?filename=trysvg_circle