

Le Standard XML

-Xpath -

Ichrak MEHREZ
(m_ichrak@hotmail.fr)

Introduction

- Deux technologies qu'il est possible d'utiliser en parallèle d'un document XML : l'API DOM et XPath.
- **DOM** ou **Document Object Model**: une technologie grâce à laquelle il est possible de lire un document XML et d'en extraire différentes informations (éléments, attributs, commentaires, etc...) afin de les exploiter.
- **XPath** est un langage permettant de sélectionner des parties d'un document XML. La syntaxe de XPath n'est pas une syntaxe XML car les expressions XPath apparaissent en général comme valeurs d'attributs de documents XML

Xpath

- XPath était au départ un langage permettant essentiellement de décrire des ensembles de nœuds dans un document XML
- La version 1.0 de XPath comprenait quelques fonctions pour la manipulation de nombres et de chaînes de caractères. L'objectif était alors de pouvoir comparer les contenus de nœuds.
- La version 2.0 de XPath a considérablement enrichi le langage. Il est devenu un langage beaucoup plus complet capable, par exemple, de manipuler des listes de nœuds et de valeurs atomiques.

Xpath

- Il n'est pas possible dans XPath de mémoriser un résultat.
- Il n'existe pas de variables propres à XPath mais une expression XPath peut référencer des variables du langage hôte.
- Le cœur de XPath est formé des expressions de chemins permettant de décrire des ensembles de nœuds d'un document XML.
- Ces expressions ressemblent aux chemins Unix pour nommer des fichiers dans une arborescence.

DONNÉES ET ENVIRONNEMENT

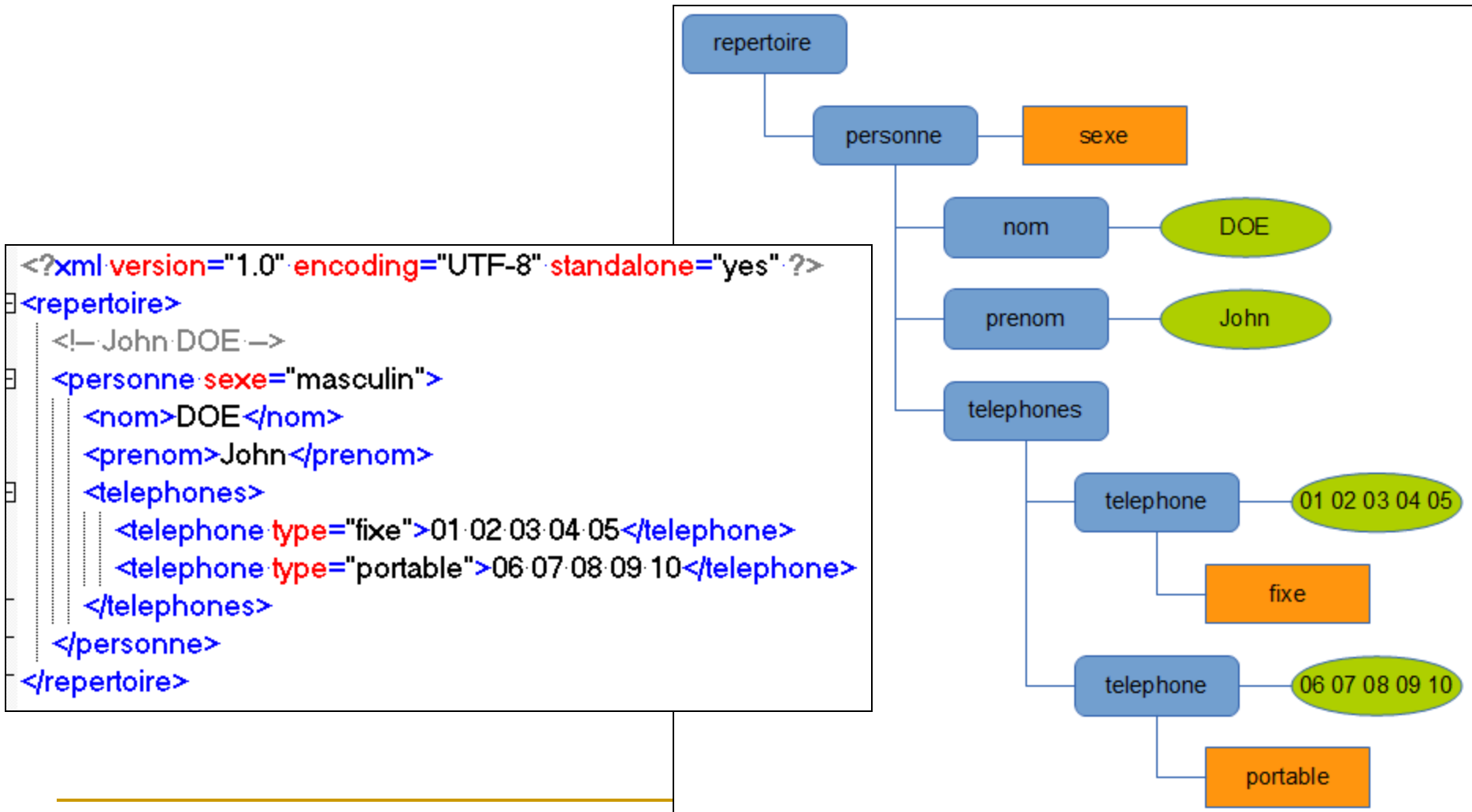
Arbre d'un document XML

- Une expression XPath est généralement évaluée par rapport à un document XML pour en sélectionner certaines parties.
- Le document XML est vu comme un arbre formé de nœuds.
- Les principaux nœuds de cet arbre sont les éléments, les attributs et le texte du document mais les commentaires et les instructions de traitement apparaissent aussi comme des nœuds.
- L'objet central de XPath est donc le nœud d'un document XML.
- XPath prend aussi en compte les contenus des éléments et les valeurs des attributs pour effectuer des sélections.

Arbre d'un document XML

- Les éléments d'un document XML sont reliés par des liens de parenté.
- Un élément est le parent d'un autre élément s'il le contient. Ces relations de parenté constituent l'arbre des éléments.
- Cette structure d'arbre est étendue à tous les constituants d'un document pour former *l'arbre du document* qui inclut les éléments et leurs contenus, les attributs, les instructions de traitement et les commentaires.
- C'est sous cette forme d'arbre que le document XML est manipulé par XPath et XSLT.

Arbre d'un document XML



Arbre d'un document XML

■ Parent

- Le **parent** d'un nœud est le nœud qui est directement au dessus de lui d'un point de vue hiérarchique.

■ Enfant

- Un nœud a pour **enfants** tous les nœuds situés un niveau en dessous dans la hiérarchie. Un nœud peut donc avoir une infinité d'enfants.

■ Descendant

- Un nœud a pour **descendants** tous les nœuds situés en dessous dans la hiérarchie. Un nœud peut donc avoir une infinité de descendants.

Arbre d'un document XML

■ Ancêtre

- Un nœud a pour **ancêtres** tous les nœuds situés en dessus dans la hiérarchie. Un nœud peut donc avoir plusieurs ancêtres.

■ Frère

- Un nœud a pour **frères** tous les nœuds situés au même niveau dans la hiérarchie. Un nœud peut donc avoir une infinité de frères.

Chemin d'accès

- Décrire le chemin emprunté dans l'arbre XML pour atteindre les données qui nous intéressent.
- Si je veux récupérer par exemple le numéro de téléphone fixe, voici le chemin à parcourir :
 - Etape 1 : nœud "**repertoire**".
 - Etape 2 : descendre au nœud enfant "**personne**".
 - Etape 3 : descendre au nœud enfant "**telephones**".
 - Etape 4 : descendre au nœud enfant "**telephone**" dont l'attribut est "**fixe**".
- L'expression XPath correspondante:

/étape1/étape2/étape3/étape4

Chemin d'accès

- Il est possible d'exprimer vos chemins de 2 manières :
 - Un **chemin absolu**: Le nœud de départ est toujours la racine de l'arbre XML. Une expression XPath utilisant un chemin absolu est facilement identifiable car elle commence par le caractère **"/**".
 - Un **chemin relatif**: Accepte quant à lui n'importe quel nœud de l'arbre XML comme point de départ. Une expression XPath utilisant un chemin relatif est facilement identifiable car elle ne commence pas par le caractère **"/**".

Composition d'une étape

- Une expression XPath est en réalité une succession d'étapes.
- Une étape est décrite par 3 éléments :
 - Un **axe**.
 - Un **nœud** ou un type de nœud.
 - Un ou plusieurs **prédicats** (facultatif).

axe::nœud [predicat][predicat]...[predicat]

Composition d'une étape

- L'**axe** va nous permettre de définir le *sens* de la recherche. Par exemple, si l'on souhaite se diriger vers un nœud enfant ou au contraire remonter vers un nœud parent voir un ancêtre.
- Le **nœud** va nous permettre d'affiner notre recherche en indiquant explicitement le **nom d'un nœud** ou le **type de nœud** dont les informations nous intéressent.
- Les **prédicats**, élément est *facultatif*, dont le nombre n'est pas limité, agissent comme un filtre et vont nous permettre de gagner en précision lors de nos recherches.

Composition d'une étape

Axe

- Le choix du sens est structuré par un vocabulaire précis:

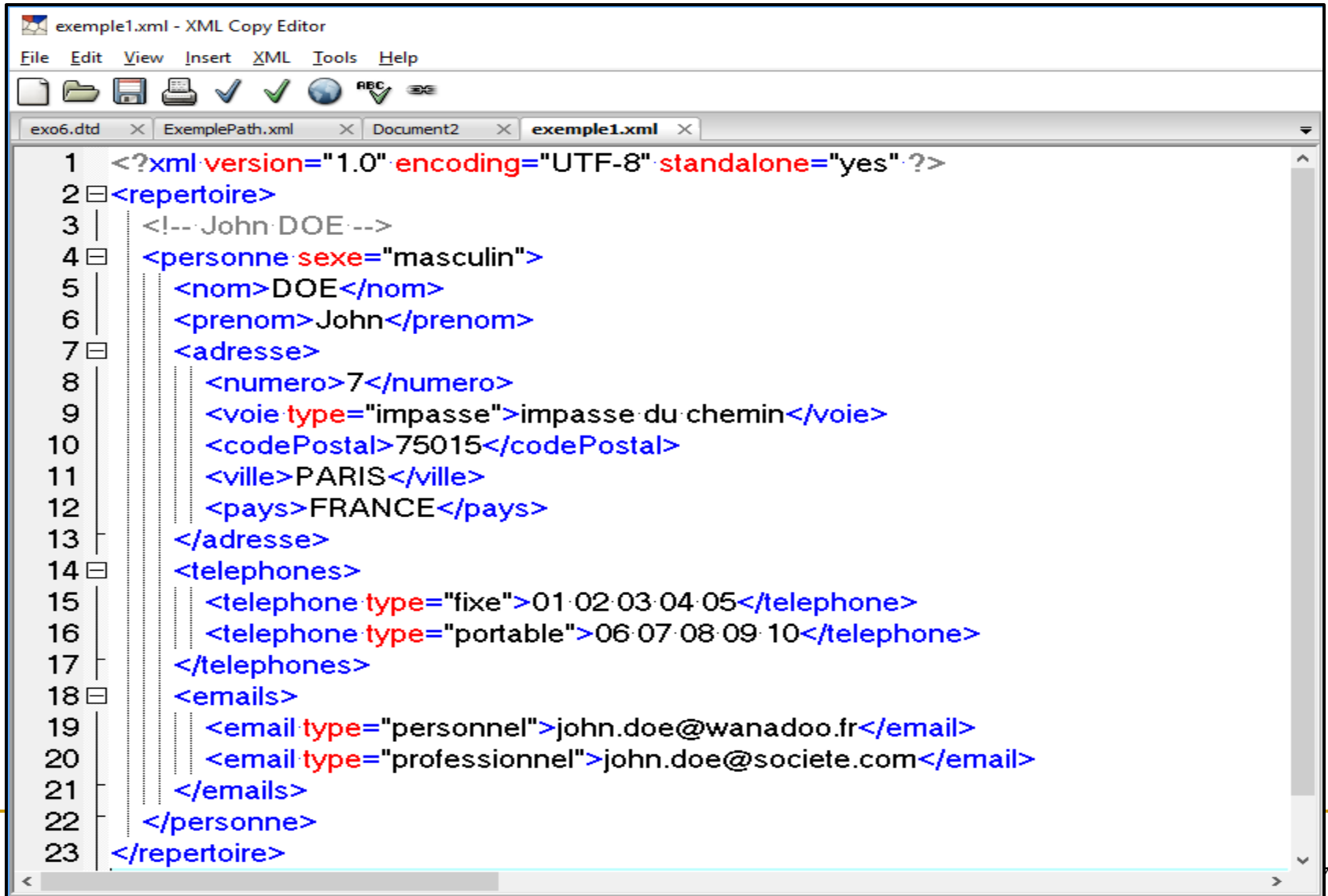
Nom de l'axe	Description: oriente la recherche vers
<i>ancestor</i>	les ancêtres du nœud courant
<i>ancestor-or-self</i>	le nœud courant et ses ancêtres
<i>attribute</i>	les attributs du nœud courant
<i>child</i>	les enfants du nœud courant
<i>descendant</i>	les descendants du nœud courant
<i>descendant-or-self</i>	le nœud courant et ses descendants
<i>following</i>	les nœuds suivant le nœud courant
<i>following-sibling</i>	les frères suivants du nœud courant
<i>parent</i>	le père du nœud courant
<i>preceding</i>	les nœuds précédant le nœud courant
<i>preceding-sibling</i>	les frères précédents du nœud courant
<i>self</i>	le nœud courant

Composition d'une étape

Noeud

Nom	Description: oriente la recherche vers
<i>nom du nœud</i>	le nœud dont le nom a explicitement été spécifié
*	tous les nœuds
<i>node()</i>	tous les types de nœuds (éléments, commentaires, attributs, etc.)
<i>text()</i>	les nœuds de type texte
<i>comment()</i>	les nœuds de type commentaire

Quelques exemples



The screenshot shows a window titled "exemple1.xml - XML Copy Editor" with a menu bar (File, Edit, View, Insert, XML, Tools, Help) and a toolbar. The editor displays an XML document with the following content:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
2 <repertoire>
3 |   <!-- John DOE -->
4 |   <personne sexe="masculin">
5 |     <nom>DOE</nom>
6 |     <prenom>John</prenom>
7 |     <adresse>
8 |       <numero>7</numero>
9 |       <voie type="impasse">impasse du chemin</voie>
10 |      <codePostal>75015</codePostal>
11 |      <ville>PARIS</ville>
12 |      <pays>FRANCE</pays>
13 |    </adresse>
14 |    <telephones>
15 |      <telephone type="fixe">01 02 03 04 05</telephone>
16 |      <telephone type="portable">06 07 08 09 10</telephone>
17 |    </telephones>
18 |    <emails>
19 |      <email type="personnel">john.doe@wanadoo.fr</email>
20 |      <email type="professionnel">john.doe@societe.com</email>
21 |    </emails>
22 |  </personne>
23 </repertoire>
```

Exemple 1

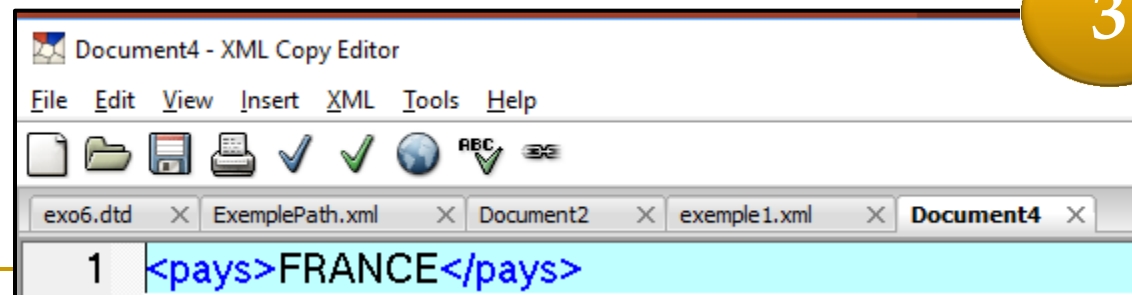
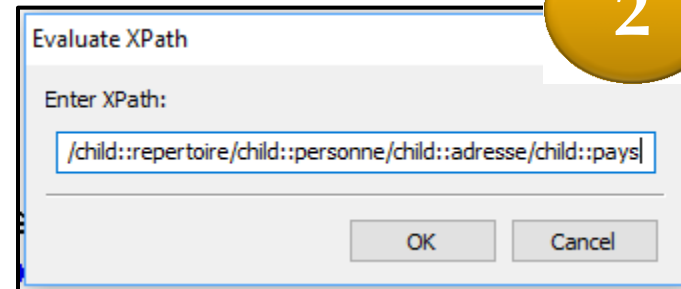
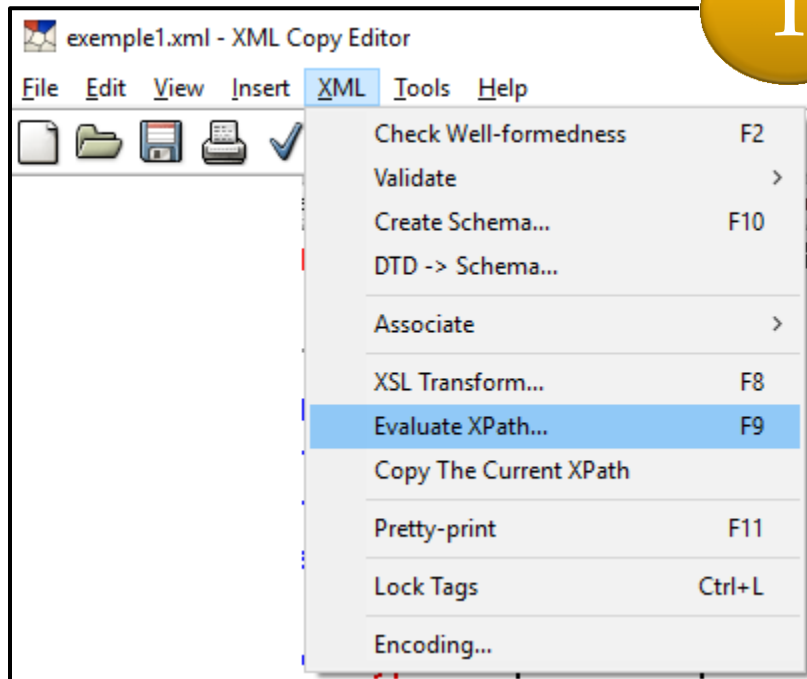
- But: récupérer le **pays de domiciliation** de John DOE.
- Les étapes:
 - Etape 1 : descendre au nœud "**repertoire**".
 - Etape 2 : descendre au nœud "**personne**".
 - Etape 3 : descendre au nœud "**adresse**".
 - Etape 4 : descendre au nœud "**pays**".

Exemple 1

- But: récupérer le **pays de domiciliation** de John DOE.
- Les étapes:
 - Etape 1 : descendre au nœud "**repertoire**".
 - Etape 2 : descendre au nœud "**personne**".
 - Etape 3 : descendre au nœud "**adresse**".
 - Etape 4 : descendre au nœud "**pays**".
- Les étapes sous la forme d'expressions XPath :
 - Etape 1 : child::repertoire.
 - Etape 2 : child::personne.
 - Etape 3 : child::adresse.
 - Etape 4 : child:pays.

/child::repertoire/child::personne/child::adresse/child::pays

Exemple 1



Exemple 2

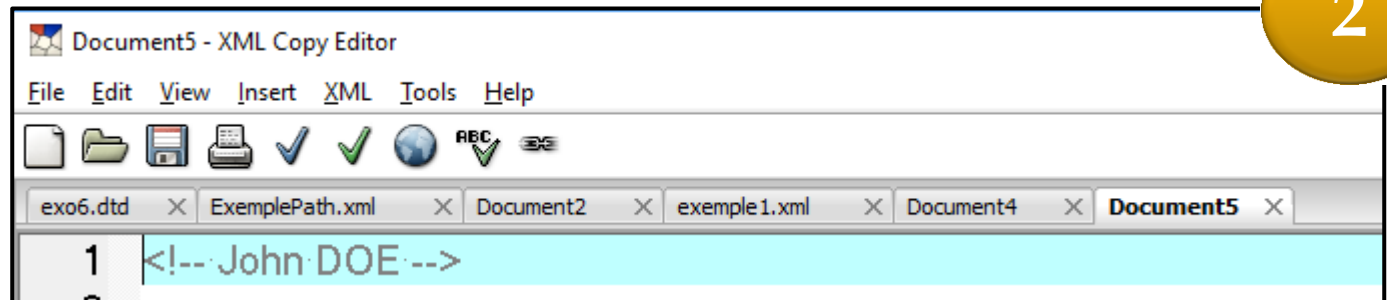
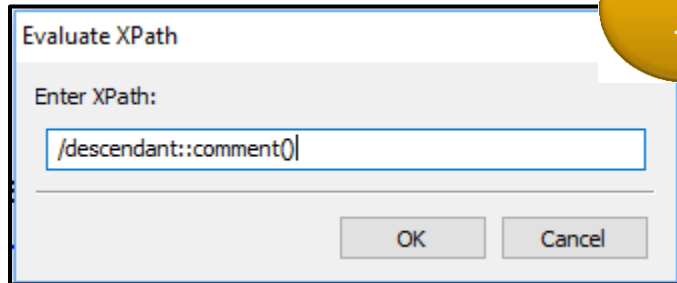
- But: **trouver tous les commentaires de notre document XML.**
- Les étapes:
 - Une seule étape: sélectionner tous les descendants du nœud **racine** qui sont des commentaires.

Exemple 2

- But: **trouver tous les commentaires de notre document XML.**
- Les étapes:
 - Une seule étape: sélectionner tous les descendants du nœud **racine** qui sont des commentaires.
- Les étapes sous la forme d'expressions XPath :
 - On sélectionne tous les descendants: `/descendant`
 - On filtre les commentaires avec l'expression **`comment()`**.

`/descendant::comment()`

Example 2



Exemple 3:

- Point de départ: le nœud "**telephones**".
- But: récupérer le **pays de domiciliation** de John DOE.
- Les étapes:
 - Etape 1 : remonter au nœud frère "**adresse**".
 - Etape 2 : descendre au nœud "**pays**".

Exemple 3:

- Point de départ: le nœud "**telephones**".
- But: récupérer le **pays de domiciliation** de John DOE.
- Les étapes:
 - ❑ Etape 1 : remonter au nœud frère "**adresse**".
 - ❑ Etape 2 : descendre au nœud "**pays**".
- Les étapes sous la forme d'expressions XPath :
 - ❑ Etape 1 : preceding-sibling::adresse.
 - ❑ Etape 2 : pays.

preceding-sibling::adresse/pays

Composition d'une étape

Prédicats

Nom du prédicat	Description
<i>attribute</i>	permet d'affiner la recherche en fonction d'un attribut
<i>count()</i>	permet de compter le nombre de nœuds
<i>last()</i>	permet de sélectionner le dernier nœud d'une liste
<i>position()</i>	permet d'affiner la recherche en fonction de la position d'un nœud

Exemple 1

- But: va être de récupérer le nœud contenant **le numéro de téléphone fixe** de John DOE.

Exemple 1

- But: va être de récupérer le nœud contenant le **numéro de téléphone fixe** de John DOE.
- Solution 1: Sélectionner tous les descendants du nœud racine et filtrer sur la valeur de l'attribut **type**.

```
/descendant::*[attribute::type="fixe"]
```

- Solution 2: préciser le chemin complet .

```
/repertoire/personne/telephones/telephone[attribute::type="fixe"]
```

Exemple 2

- But: sélectionner les numéros de téléphones qui ne sont pas des numéros de téléphones fixes.
- Solution 1: Remplacer dans notre expression précédente l'opérateur d'égalité "=" par l'opérateur de non égalité "!=" :

```
/repertoire/personne/telephones/telephone[attribute::type!="fixe"]
```

- Solution 2: utiliser la fonction **not()** :

```
/repertoire/personne/telephones/telephone[not(attribute::type="fixe")]
```

Exemple 3

- But: sélectionner le **premier numéro de téléphone** de John DOE.
- Les étapes:
 - Etape 1 : descendre au nœud "**repertoire**".
 - Etape 2 : descendre au nœud "**personne**".
 - Etape 3 : descendre au nœud "**telephones**".
 - Etape 4 : sélectionner le premier nœud "**telephone**".

Exemple 3

- But: sélectionner le **premier numéro de téléphone** de John DOE.
- Les étapes:
 - Etape 1 : descendre au nœud "**repertoire**".
 - Etape 2 : descendre au nœud "**personne**".
 - Etape 3 : descendre au nœud "**telephones**".
 - Etape 4 : sélectionner le premier nœud "**telephone**".
- Les étapes sous la forme d'expressions XPath :
 - Etape 1 : repertoire.
 - Etape 2 : personne.
 - Etape 3 : telephones.
 - Etape 4 : telephone[**position()**=1].

/repertoire/personne/telephones/telephone[position()=1]

Exercice

- Sélectionner tous les nœuds descendants du deuxième nœud "**personne**".
- Sélectionner le nœud "**personne**" correspondant à l'individu ayant au moins 2 numéros de téléphone.
- Sélectionner tous les nœuds "**personne**".
- Sélectionner le deuxième nœud "**personne**" dont le pays de domiciliation est la **France** .
- Sélectionner tous les nœuds "**personne**" de sexe **masculin** le pays de domiciliation est les **Etats-Unis**.

Solution

- `/repertoire/personne[position()=2]/descendant::*`
- `/repertoire/personne[count(telephones/telephone) > 1]`
- `/repertoire/personne`
- `/repertoire/personne[adresse/pays="FRANCE"][position()=2]`
- `/repertoire/personne[@sexe="masculin"][adresse/pays="USA"]`