

Documents structurés

Cours 4

Nassim ZELLAL

XPath - 1

Les expressions XPath

- Le langage XPath s'appuie sur des **expressions** désignant un ou plusieurs nœuds, en exprimant des chemins de localisation dans l'arbre d'un document XML.
- Une **expression XPath** représente donc un chemin de localisation pour naviguer dans l'arbre d'un document XML et sélectionner un ensemble de nœuds.
- Tester les expressions XPath avec « Exchanger XML Editor » :
 - ❑ <https://www.softpedia.com/get/Internet/WEB-Design/HTML-Editors/Exchanger-XML-Editor.shtml>

Le nœud racine (root node)


- Dans l'expression XPath « **/livre/chapitre** », la première barre oblique **/** (slash) indique que la recherche commence à partir du **nœud racine** (ou nœud de document) de l'arbre du document XML.
- Dans l'exemple ci-dessous, l'enfant direct du **nœud racine** **/** est le nœud d'élément racine (ou élément racine ou élément de document) **<livre>**. Ce dernier est le seul enfant du **nœud racine**.
- Le **nœud racine** **/** contient tout le document XML et inclut donc le nœud d'élément racine **<livre>**.
- Le **nœud racine** représente donc tout le document.
- L'évaluation de l'expression XPath composée d'un **/** retourne le document XML tout entier.

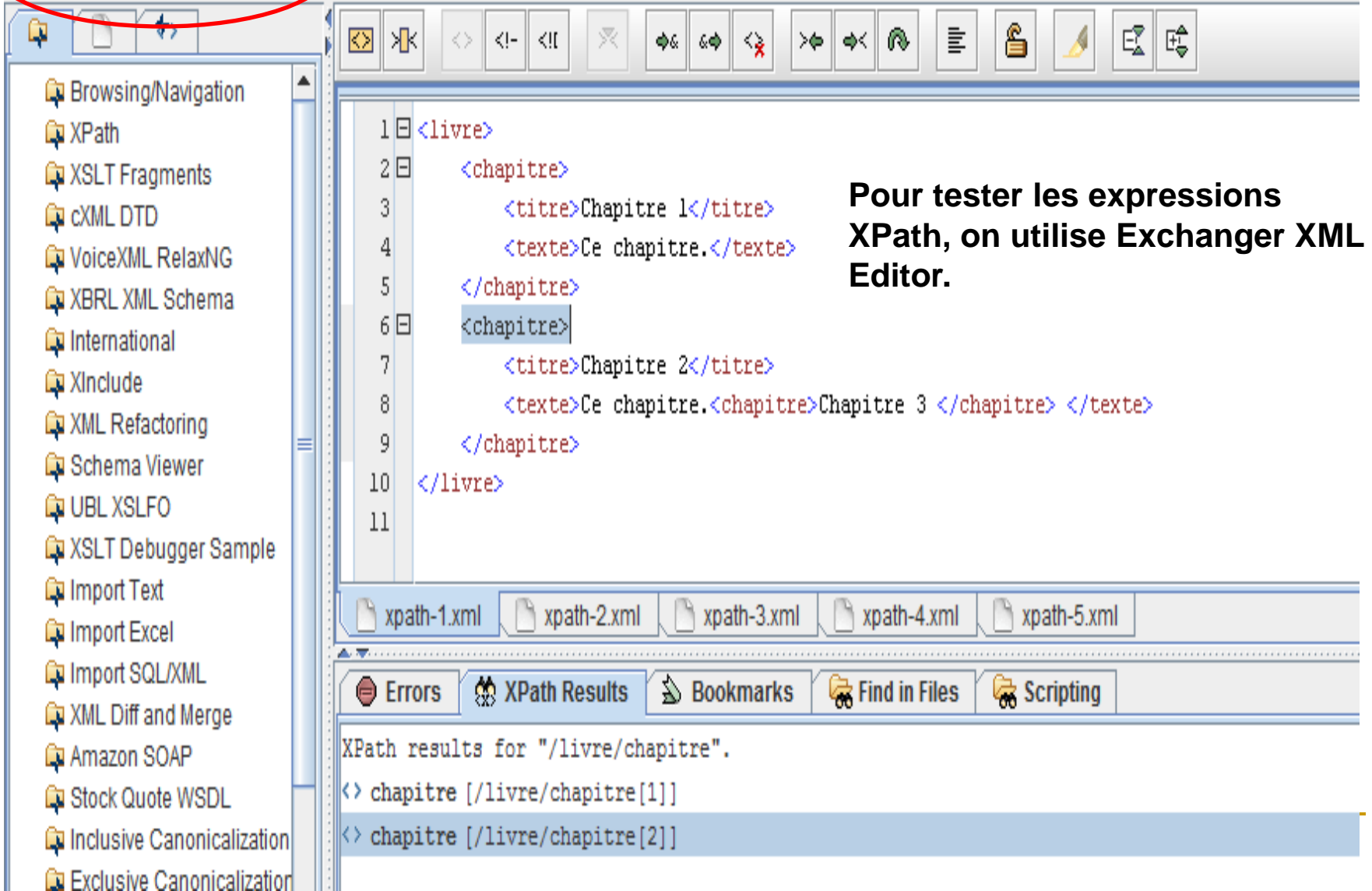
- **<livre>**
- **<chapitre>**
- **<titre>Chapitre 1</titre>**
- **<texte>Ce chapitre.</texte>**
- **</chapitre>**
- **<chapitre>**
- **<titre>Chapitre 2</titre>**
- **<texte>Ce chapitre.**
- **<chapitre>Chapitre 3<chapitre/>**
- **</texte>**
- **</chapitre>**
- **</livre>**

Le nœud d'élément

- Localiser les nœuds « chapitre » dans le document XML « xpath-1.xml » avec l'expression : `//livre/chapitre`
- `<livre>`
- `<chapitre>`
- `<titre>Chapitre 1</titre>`
- `<texte>Ce chapitre.</texte>`
- `</chapitre>`
- `<chapitre>`
- `<titre>Chapitre 2</titre>`
- `<texte>Ce chapitre.`
- `<chapitre>Chapitre 3<chapitre/>`
- `</texte>`
- `</chapitre>`
- `</livre>`
- Voir xpath-1.xml

Le nœud d'élément

XPath:  /livre/chapitre



The screenshot displays the Exchanger XML Editor interface. On the left is a sidebar with a tree view of XML-related tools, including Browsing/Navigation, XPath, XSLT Fragments, cXML DTD, VoiceXML RelaxNG, XBRL XML Schema, International, XInclude, XML Refactoring, Schema Viewer, UBL XSLFO, XSLT Debugger Sample, Import Text, Import Excel, Import SQL/XML, XML Diff and Merge, Amazon SOAP, Stock Quote WSDL, Inclusive Canonicalization, and Exclusive Canonicalization. The main editor area shows an XML document with the following structure:

```
1 <livre>
2   <chapitre>
3     <titre>Chapitre 1</titre>
4     <texte>Ce chapitre.</texte>
5   </chapitre>
6   <chapitre>
7     <titre>Chapitre 2</titre>
8     <texte>Ce chapitre.<chapitre>Chapitre 3 </chapitre> </texte>
9   </chapitre>
10 </livre>
11
```

Below the XML editor is a tabbed interface with tabs for Errors, XPath Results, Bookmarks, Find in Files, and Scripting. The XPath Results tab is active, showing the results for the XPath expression `"/livre/chapitre"`:

```
<> chapitre [/livre/chapitre[1]]
<> chapitre [/livre/chapitre[2]]
```

On the right side of the editor, there is a text box that reads: **Pour tester les expressions XPath, on utilise Exchanger XML Editor.**

Le nœud d'élément

- Autre syntaxe : **//** parcourir **récurivement** tous les nœuds **descendants** à partir du nœud d'élément courant (ou contextuel).
- Exemple : **//chapitre** permet d'extraire (sélectionner) tous les nœuds d'élément « chapitre ».
- `<livre>`
- **`<chapitre>`**
- `<titre>Chapitre 1</titre>`
- `<texte>Ce chapitre.</texte>`
- **`</chapitre>`**
- **`<chapitre>`**
- `<titre>Chapitre 2</titre>`
- `<texte>Ce chapitre.<chapitre>Chapitre 3 </chapitre>`
- `</texte>`
- **`</chapitre>`**
- `</livre>`
- **Voir xpath-1.xml**

Le nœud d'élément

The screenshot shows an XML editor interface. At the top, the XPath bar displays `XPath: //chapitre`, which is circled in red. Below this is a toolbar with various XML editing icons. On the left, a sidebar lists several tools including Browsing/Navigation, XPath, XSLT Fragments, cXML DTD, VoiceXML RelaxNG, XBRL XML Schema, International, XInclude, XML Refactoring, Schema Viewer, UBL XSLFO, XSLT Debugger Sample, Import Text, Import Excel, Import SQL/XML, XML Diff and Merge, Amazon SOAP, Stock Quote WSDL, Inclusive Canonicalization, and Exclusive Canonicalization. The main editor area displays an XML document with the following structure:

```
1 <livre>
2   <chapitre>
3     <titre>Chapitre 1</titre>
4     <texte>Ce chapitre.</texte>
5   </chapitre>
6   <chapitre>
7     <titre>Chapitre 2</titre>
8     <texte>Ce chapitre.<chapitre>Chapitre 3 </chapitre> </texte>
9   </chapitre>
10 </livre>
11
```


Below the editor, there are tabs for `xpath-1.xml`, `xpath-2.xml`, `xpath-3.xml`, `xpath-4.xml`, and `xpath-5.xml`. At the bottom, a panel contains tabs for Errors, XPath Results, Bookmarks, Find in Files, and Scripting. The XPath Results tab is active, showing the following results for the expression `"//chapitre"`:

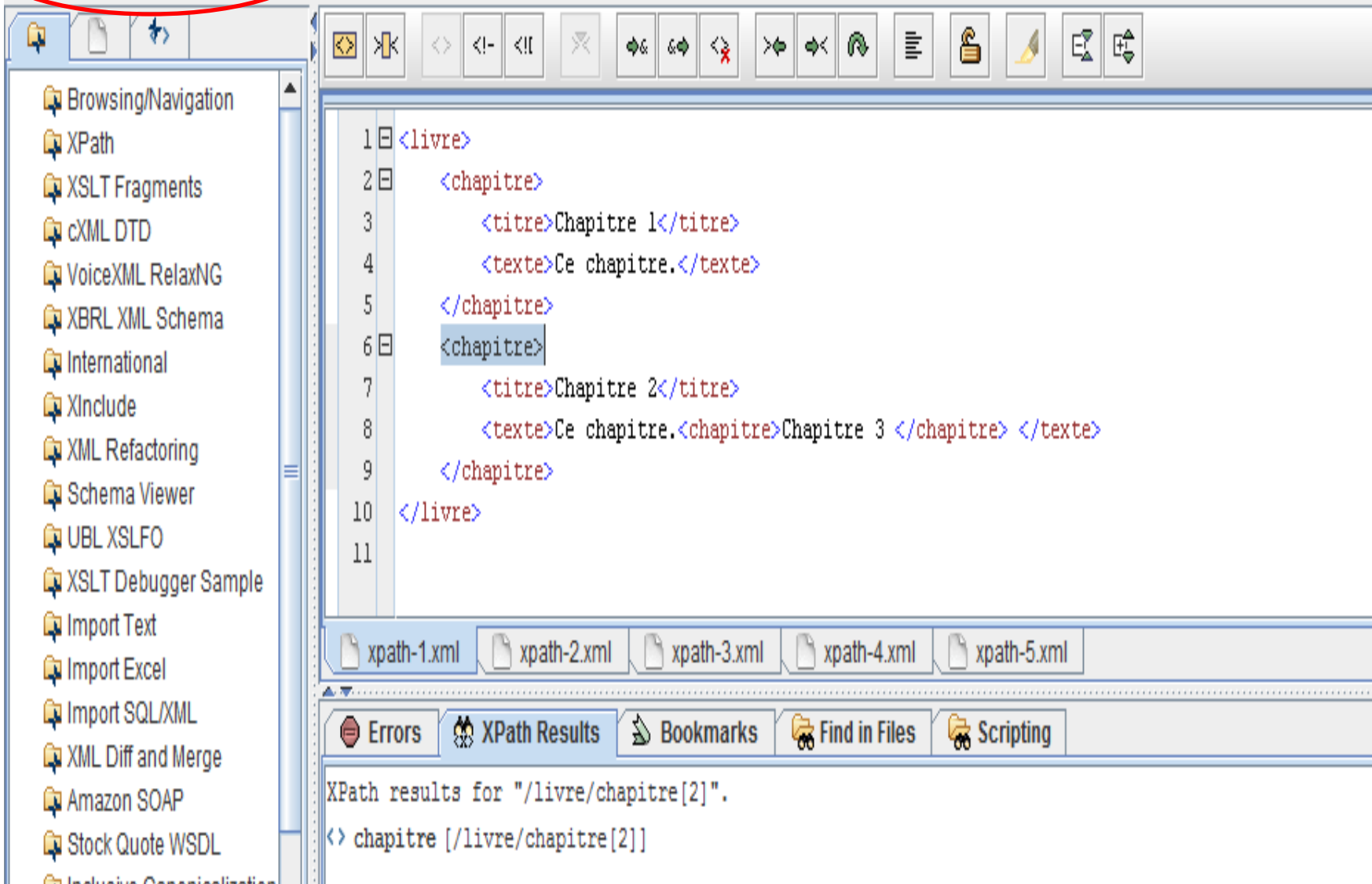
```
XPath results for "//chapitre".
<> chapitre [/livre/chapitre[1]]
<> chapitre [/livre/chapitre[2]]
<> chapitre [/livre/chapitre[2]/texte/chapitre]
```


Le nœud d'élément (position)

- Pour orienter la recherche sur la position d'un nœud, on utilise les crochets [] : /livre/chapitre[2]
- <livre>
- <chapitre>
- <titre>Chapitre 1</titre>
- <texte>Ce chapitre.</texte>
- </chapitre>
- <chapitre>
- <titre>Chapitre 2</titre>
- <texte>Ce chapitre.<chapitre>Chapitre 3 </chapitre>
- </texte>
- </chapitre>
- </livre>
- Voir xpath-1.xml

Le nœud d'élément (position)

XPath:  /livre/chapitre[2]



The screenshot shows an XML editor interface. On the left is a sidebar with a tree view containing various XML-related items like 'Browsing/Navigation', 'XPath', 'XSLT Fragments', etc. The main area displays an XML document with the following structure:

```
1 <livre>
2   <chapitre>
3     <titre>Chapitre 1</titre>
4     <text>Ce chapitre.</text>
5   </chapitre>
6   <chapitre>
7     <titre>Chapitre 2</titre>
8     <text>Ce chapitre.<chapitre>Chapitre 3 </chapitre> </text>
9   </chapitre>
10 </livre>
11
```

Below the XML view is a tabbed interface with tabs for 'xpath-1.xml', 'xpath-2.xml', 'xpath-3.xml', 'xpath-4.xml', and 'xpath-5.xml'. At the bottom, there is a panel with tabs for 'Errors', 'XPath Results', 'Bookmarks', 'Find in Files', and 'Scripting'. The 'XPath Results' tab is active, showing the results for the XPath query `/livre/chapitre[2]`:

```
XPath results for "/livre/chapitre[2]".
<> chapitre [/livre/chapitre[2]]
```

Le nœud d'élément (position)

- Question : comment retrouver Chapitre 3 ?

Le nœud d'élément (position)

- Réponse :
- **//chapitre[2]/texte/chapitre**
- **/livre/chapitre/texte/chapitre**

Le nœud d'élément (position)

The screenshot shows an XML editor interface. On the left is a sidebar with a tree view containing various XML-related tools and templates, such as 'Browsing/Navigation', 'XPath', 'XSLT Fragments', 'cXML DTD', 'VoiceXML RelaxNG', 'XBRL XML Schema', 'International', 'XInclude', 'XML Refactoring', 'Schema Viewer', 'UBL XSLFO', 'XSLT Debugger Sample', 'Import Text', 'Import Excel', 'Import SQL/XML', 'XML Diff and Merge', 'Amazon SOAP', and 'Stock Quote WSDL'.

The main editor area displays an XML document with the following structure:

```
1 <livre>
2   <chapitre>
3     <titre>Chapitre 1</titre>
4     <texte>Ce chapitre.</texte>
5   </chapitre>
6   <chapitre>
7     <titre>Chapitre 2</titre>
8     <texte>Ce chapitre.<chapitre>Chapitre 3 </chapitre> </texte>
9   </chapitre>
10 </livre>
11
```

The text node on line 8 contains an embedded chapter element. The editor's status bar at the bottom shows several tabs: 'xpath-1.xml', 'xpath-2.xml', 'xpath-3.xml', 'xpath-4.xml', and 'xpath-5.xml'. Below these is a row of buttons: 'Errors', 'XPath Results', 'Bookmarks', 'Find in Files', and 'Scripting'. The 'XPath Results' tab is active, displaying the following text:

XPath results for "//chapitre[2]/texte/chapitre".

```
<> chapitre [/livre/chapitre[2]/texte/chapitre]
```

Le nœud d'attribut

- Pour localiser un nœud d'attribut, on emploie les crochets (Voir aussi les « Prédicats », slides 28, 29 et 30) :
- `//personne[@prenom="Jack"]`
- **Pour tout localiser ==> `//personne[@prenom]`**
- `<liste>`
- `< personne prenom ="Jack" nom= "Nolan" />`
- `< personne prenom="Jean" nom ="Palvadeau" />`
- `< personne prenom="Billy" nom = "Madison" />`
- `< personne prenom="John" nom ="Woo" date_de_naissance="10-12-1990"/>`
- `</liste>`
- **Voir xpath-2.xml**

Le nœud d'attribut

```
XPath results for "//personne[@prenom="Jack"]".  
<> personne [/liste/personne[1]]
```

Le nœud d'attribut

```
XPath results for "//personne[@prenom]".
```

```
<> personne [/liste/personne[1]]
```

```
<> personne [/liste/personne[2]]
```

```
<> personne [/liste/personne[3]]
```

```
<> personne [/liste/personne[4]]
```


Le nœud d'attribut

- /nœud_courant/**@**type (sélectionne l'attribut type du nœud nœud d'élément courant)
- Tester avec :
- **/liste/personne/**@nom

Le nœud d'attribut

XPath results for `"/liste/personne/@nom"`.

@ nom [/liste/personne[1]/@nom]

@ nom [/liste/personne[2]/@nom]

@ nom [/liste/personne[3]/@nom]

@ nom [/liste/personne[4]/@nom]

Le nœud d'attribut

- Question : quelle autre expression peut-on utiliser pour obtenir le même résultat ?

Le nœud d'attribut

- Réponse :
- **//personne/@nom**

Wildcard (joker) *

- Le caractère astérisque (**Wildcard**) appelé aussi joker * correspond à n'importe quel élément contenu dans le nœud d'élément courant. Cela correspond donc aux nœuds d'élément sans tenir compte de leur nom.
- Exemple 1:
- /AAA/BBB/* (sélectionne tous les enfants directs du nœud d'élément courant /AAA/BBB).
- Exemple 2:
- /AAA/BBB/HHH/@* (sélectionne tous les attributs (ou nœuds d'attribut) du nœud courant /AAA/BBB/HHH).
- Exemple 3:
- /AAA/BBB//* (sélectionne tous les descendants du nœud d'élément courant /AAA/BBB).
- **Voir xpath-3.xml**

Wildcard (joker) *

The screenshot displays the XMLSpy application interface. The 'XPath' field in the top toolbar is circled in red and contains the query `/AAA/BBB/*`. The main editor shows an XML document with the following structure:

```
1 <AAA>
2   <BBB/>
3   <CCC/>
4   <BBB>
5     <HHH att="1" />
6     <HHH att="2" />
7     <HHH fromage="Gouda" />
8     <HHH>
9       <TTT/>
10    </HHH>
11    <HHH />
12  </BBB>
13  <BCC>
14    <BCD>
15      <HHH fromage="Cantal" />
16      <CDB />
17    </BCD>
18  </BCC>
19  <DDD>
20    <EEE />
```

The 'XPath Results' tab at the bottom shows the results for the query `"/AAA/BBB/*"`:

```
<> HHH [/AAA/BBB[2]/HHH[1]]
<> HHH [/AAA/BBB[2]/HHH[2]]
<> HHH [/AAA/BBB[2]/HHH[3]]
<> HHH [/AAA/BBB[2]/HHH[4]]
<> HHH [/AAA/BBB[2]/HHH[5]]
<> DDD [/AAA/BBB[3]/DDD]
```

The 'Document Properties' panel on the left indicates the document is UTF-8 encoded and has no schema defined.

Wildcard (joker) *

The screenshot displays the XMLSpy application interface. The 'XPath' field at the top is highlighted with a red circle and contains the query `/AAA/BBB/HHH/@*`. The main editor shows an XML document with the following structure:

```
1 <AAA>
2   <BBB/>
3   <CCC/>
4   <BBB>
5     <HHH att="1" />
6     <HHH att="2" />
7     <HHH fromage="Gouda" />
8   </BBB>
9   <TTT/>
10  </HHH>
11  <HHH />
12 </BBB>
13 <BCC>
14   <BCD>
15     <HHH fromage="Cantal" />
16     <CDB />
17   </BCD>
18 </BCC>
19 <DDD>
20   <EEE />
```

The 'XPath Results' tab at the bottom shows the results for the query `/AAA/BBB/HHH/@*`:

```
XPath results for "/AAA/BBB/HHH/@*".
@ att [/AAA/BBB[2]/HHH[1]/@att]
@ att [/AAA/BBB[2]/HHH[2]/@att]
@ fromage [/AAA/BBB[2]/HHH[3]/@fromage]
```

Wildcard (joker) *

The screenshot displays the XMLSpy application interface. At the top, the menu bar includes File, Edit, View, Project, XML, Schema, Transform, Security, Tools, and Help. Below the menu is a toolbar with various icons. The main window is divided into several panes. On the left, the 'Browsing/Navigation' pane is highlighted with a red circle and contains a list of XPath-related functions, including 'XPath', 'XPath Fragments', 'cXML DTD', 'VoiceXML RelaxNG', 'XBRL XML Schema', 'International', 'XInclude', 'XML Refactoring', 'Schema Viewer', 'UBL XSLFO', 'XSLT Debugger Sample', 'Import Text', 'Import Excel', 'Import SQL/XML', 'XML Diff and Merge', 'Amazon SOAP', 'Stock Quote WSDL', 'Inclusive Canonicalization', 'Exclusive Canonicalization', 'Envelope XML Signature', 'Detached XML Signature', 'XQuery', and 'SVG'. The 'XPath' item is selected. The central pane shows an XML document with the following structure:

```
1 <AAA>
2   <BBB/>
3   <CCC/>
4   <BBB>
5     <HHH att="1"/>
6     <HHH att="2"/>
7     <HHH fromage="Gouda" />
8   <HHH>
9     <TTT/>
10  </HHH>
11  <HHH />
12 </BBB>
13 <BCC>
14   <BCD>
15     <HHH fromage="Cantal" />
16   <CDB />
```

At the bottom, the 'XPath Results' pane shows the results of the XPath query `/AAA/BBB//*`. The results are listed as follows:

```
XPath results for "/AAA/BBB//*".
<> HHH [/AAA/BBB[2]/HHH[1]]
<> HHH [/AAA/BBB[2]/HHH[2]]
<> HHH [/AAA/BBB[2]/HHH[3]]
<> HHH [/AAA/BBB[2]/HHH[4]]
<> TTT [/AAA/BBB[2]/HHH[4]/TTT]
<> HHH [/AAA/BBB[2]/HHH[5]]
<> DDD [/AAA/BBB[3]/DDD]
<> CCC [/AAA/BBB[3]/DDD/CCC]
<> DDD [/AAA/BBB[3]/DDD/CCC/DDD]
<> EEE [/AAA/BBB[3]/DDD/CCC/EEE]
```


Les nœuds attributs (opérateurs logiques)

- `//personne[@prenom="Jack"] or //personne[@prenom="Nolan"]`
- Renvoie : true
- `//personne[@prenom="Jack"] and //personne[@prenom="Nolan"]`
- Renvoie : false

- `<liste>`
- `< personne prenom ="Jack" nom= "Nolan" />`
- `< personne prenom="Jean" nom ="Palvadeau" />`
- `< personne prenom="Billy" nom = "Madison" />`
- `< personne prenom="John" nom ="Woo" date_de_naissance="10-12-1990"/>`
- `</liste>`

- [Voir xpath-2.xml](#)

Le nœud d'attribut (opérateurs booléens)

```
XPath results for "//personne[@prenom="Jack"] or //personne[@prenom="Nolan"]  
true
```

Les nœuds attributs (opérateurs logiques)

- `//personne[@prenom and @date_de_naissance]`
- `<liste>`
- `< personne prenom ="Jack" nom= "Nolan" />`
- `< personne prenom="Jean" nom ="Palvadeau" />`
- `< personne prenom="Billy" nom = "Madison" />`
- `< personne prenom="John" nom="Woo" date_de_naissance="10-12-1990"/>`
- `</liste>`
- [Voir xpath-2.xml](#)

File Edit View Project XML Schema Transform Security Tools Help



XPath: //personne[@prenom and @date_de_naissance]



- Browsing/Navigation
- XPath
- XSLT Fragments
- cXML DTD
- VoiceXML RelaxNG
- XBRL XML Schema
- International
- XInclude
- XML Refactoring
- Schema Viewer
- UBL XSLFO
- XSLT Debugger Sample
- Import Text
- Import Excel
- Import SQL/XML
- XML Diff and Merge
- Amazon SOAP
- Stock Quote WSDL
- Inclusive Canonicalization
- Exclusive Canonicalization
- Envelope XML Signature
- Detached XML Signature
- XQuery



```
1 <liste>
2 <personne prenom="Jack" nom="Nolan" />
3 <personne prenom="Jean" nom="Palvadeau" />
4 <personne prenom="Billy" nom="Madison" />
5 <personne prenom="John" nom="Woo" date_de_naissance="10-12-1990" />
6 </liste>
7
```

xpath-1.xml xpath-2.xml xpath-3.xml xpath-4.xml xpath-5.xml

Errors XPath Results Bookmarks Find in Files Scripting

XPath results for "//personne[@prenom and @date_de_naissance]".

<> personne [/liste/personne[4]]

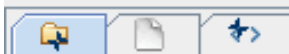
Les nœuds attributs (opérateurs logiques)

- `//personne[not(contains(@prenom, 'Billy'))]`
- `<liste>`
- `< personne prenom ="Jack" nom= "Nolan" />`
- `< personne prenom="Jean" nom ="Palvadeau" />`
- `< personne prenom="Billy" nom = "Madison" />`
- `< personne prenom="John" nom ="Woo" date_de_naissance="10-12-1990"/>`
- `</liste>`
- [Voir xpath-2.xml](#)

File Edit View Project XML Schema Transform Security Tools Help



XPath: `//personne[not(contains(@prenom, 'Billy'))]`



- Browsing/Navigation
- XPath
- XSLT Fragments
- cXML DTD
- VoiceXML RelaxNG
- XBRL XML Schema
- International
- XInclude
- XML Refactoring
- Schema Viewer
- UBL XSLFO
- XSLT Debugger Sample
- Import Text
- Import Excel
- Import SQL/XML
- XML Diff and Merge
- Amazon SOAP
- Stock Quote WSDL
- Inclusive Canonicalization
- Exclusive Canonicalization
- Envelope XML Signature
- Detached XML Signature
- XQuery
- SVG



```
1 <liste>
2 <personne prenom="Jack" nom="Nolan" />
3 <personne prenom="Jean" nom="Palvadeau" />
4 <personne prenom="Billy" nom="Madison" />
5 <personne prenom="John" nom="Woo" date_de_naissance="10-12-1990" />
6 </liste>
7
```

xpath-1.xml xpath-2.xml xpath-3.xml xpath-4.xml xpath-5.xml

Errors XPath Results Bookmarks Find in Files Scripting

XPath results for `"//personne[not(contains(@prenom, 'Billy'))]"`.

```
<> personne [/liste/personne[1]]
<> personne [/liste/personne[2]]
<> personne [/liste/personne[4]]
```

Les fonctions

- `node()` → `//livre/chapitre/texte/node()`
- `position()` → `//nom[position()=3]`
- `last()` → `//nom[last()] == //nom[position()=last()]`
- `text()` → `//Capot/text()`
- `count()` → `count(//nom)`
- `name()` → `//*[name()='BBB']`
- `start-with()` → `//*[starts-with(name(),'B')]`
- `string-length()` → `string-length(//Capot)`
- `contains()` → `contains(//Capot/text(), 'peu')` **renvoie un booléen**
- `concat()` → `concat(/Voiture/Carrosserie/Capot/text(),
'',//Allumage/text())`
- `comment()` → `//comment()`

Les fonctions

XPath: `concat(/Voiture/Carrosserie/Capot/text(),' ',//Allumage/text())`

The screenshot shows an XML editor interface. On the left is a sidebar with a tree view containing various XML-related tools like 'Browsing/Navigation', 'XPath', 'XSLT Fragments', etc. The main area displays an XML document with the following content:

```
1 <?xml version='1.0' ?>
2 <!DOCTYPE Voiture SYSTEM "Voiture.dtd">
3 <Voiture marque="Renault" modèle="Safrane">
4   <Carrosserie couleur="rouge">
5     <Capot>Un peu cabossé</Capot>
6   </Carrosserie>
7   <Moteur>
8     <Cylindres />
9     <Allumage>Défectueux</Allumage>
10  </Moteur>
11  <Transmission type="automatique" nb_vitesses="5">
12    <Boîte />
```

Below the XML document is a tabbed interface with tabs for 'xpath-1.xml' through 'xpath-5.xml'. At the bottom, there is a panel with tabs for 'Errors', 'XPath Results', 'Bookmarks', 'Find in Files', and 'Scripting'. The 'XPath Results' tab is active, showing the results of the XPath query:

XPath results for "concat(/Voiture/Carrosserie/Capot/text(),' ',//Allumage/text())".
Un peu cabossé Défectueux

Les prédicats

- L'ensemble des nœuds obtenus par un chemin XPath peut être filtré à l'aide d'un prédicat entre crochets **[]**. Cela permet de raffiner notre sélection.
- Exemple :
- `//atome[masse < 40]`
- `//*[string-length(name ()) = 5]` (donne les éléments qui ont un nom dont le nombre de caractères est exactement 5)
- **Voir xpath-4.xml**

Les prédicats

The screenshot shows an XML editor interface. At the top, the XPath query `//atome[masse < 40]` is entered in the top bar and is circled in red. The main editor displays an XML document with the following structure:

```
1 <?xml version="1.0" encoding="iso-8859-1" ?>
2 <classification_atomique>
3   <!--commentaire-->
4   <famille type="gaz rare">
5     <atome>
6       <nom>hélium</nom>
7       <symbole>He<autre>autre</autre></symbole>
8       <numero>2</numero>
9       <masse>4</masse>
10    </atome>
11    <atome>
12      <nom>néon</nom>
```

Below the XML editor, there are tabs for `xpath-1.xml`, `xpath-2.xml`, `xpath-3.xml`, `xpath-4.xml`, and `xpath-5.xml`. At the bottom, there are tabs for `Errors`, `XPath Results`, `Bookmarks`, `Find in Files`, and `Scripting`. The `XPath Results` tab is active, showing the results of the query:

```
XPath results for "//atome[masse < 40]".
<> atome [/classification_atomique/famille/atome[1]]
<> atome [/classification_atomique/famille/atome[2]]
```

Les prédicats

The screenshot shows an XML editor interface. At the top, the XPath query `//*[string-length(name ()) = 5]` is entered in the XPath field, which is circled in red. Below the query field is a toolbar with various icons. The main editor area displays an XML document with the following content:

```
1 <?xml version="1.0" encoding="iso-8859-1" ?>
2 <classification_atomique>
3 <!--commentaire-->
4 <famille type="gaz rare">
5   <atome>
6     <nom>hélium</nom>
7     <symbole>He<autre>autre</autre></symbole>
8     <numero>2</numero>
9     <masse>4</masse>
10   </atome>
11   <atome>
12     <nom>néon</nom>
```

Below the XML document is a tabbed interface with tabs for `xpath-1.xml`, `xpath-2.xml`, `xpath-3.xml`, `xpath-4.xml`, and `xpath-5.xml`. The `xpath-4.xml` tab is selected. Below the tabs is a section with tabs for `Errors`, `XPath Results`, `Bookmarks`, `Find in Files`, and `Scripting`. The `XPath Results` tab is selected, showing the results of the XPath query:

XPath results for `"//*[string-length(name ()) = 5]"`.

- `<> atome [/classification_atomique/famille/atome[1]]`
- `<> autre [/classification_atomique/famille/atome[1]/symbole/autre]`
- `<> masse [/classification_atomique/famille/atome[1]/masse]`
- `<> atome [/classification_atomique/famille/atome[2]]`
- `<> masse [/classification_atomique/famille/atome[2]/masse]`
- `<> atome [/classification_atomique/famille/atome[3]]`
- `<> masse [/classification_atomique/famille/atome[3]/masse]`

On the left side of the editor, there is a sidebar with a tree view containing various XML-related items, including `Browsing/Navigation`, `XPath`, `XSLT Fragments`, `cXML DTD`, `VoiceXML RelaxNG`, `XBRL XML Schema`, `International`, `XInclude`, `XML Refactoring`, `Schema Viewer`, `UBL XSLFO`, `XSLT Debugger Sample`, `Import Text`, `Import Excel`, `Import SQL/XML`, `XML Diff and Merge`, `Amazon SOAP`, `Stock Quote WSDL`, `Inclusive Canonicalization`, `Exclusive Canonicalization`, and `Envelope XML Signature`. At the bottom left, there is a `Document Properties` section showing the `Type` and `Encoding` of the document.

Exercice

- 1-Extraire tous les nœuds de texte du document "xpath-1.xml".
 - Proposer deux solutions.
- 2-Extraire tous les nœuds d'élément "HHH" du document "xpath-3.xml".
- 3-Extraire tous les nœuds d'attribut du document "xpath-3.xml".
- 4-Extraire tous les nœuds d'élément "HHH", qui possèdent un nœud d'attribut du document "xpath-3.xml".
- 5-Extraire le nœud d'élément contenant un nœud d'attribut avec la valeur « Cantal » du document "xpath-3.xml".
- 6-Extraire les nœuds d'élément "atome" ayant une masse strictement inférieure à "20" du document "xpath-4.xml".
- 7-Extraire les nœuds d'élément vides du document "xpath-5.xml".