

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>
 - ❑ <https://code.google.com/archive/p/exchangerxml/downloads>
-


Le nœud racine (root node)

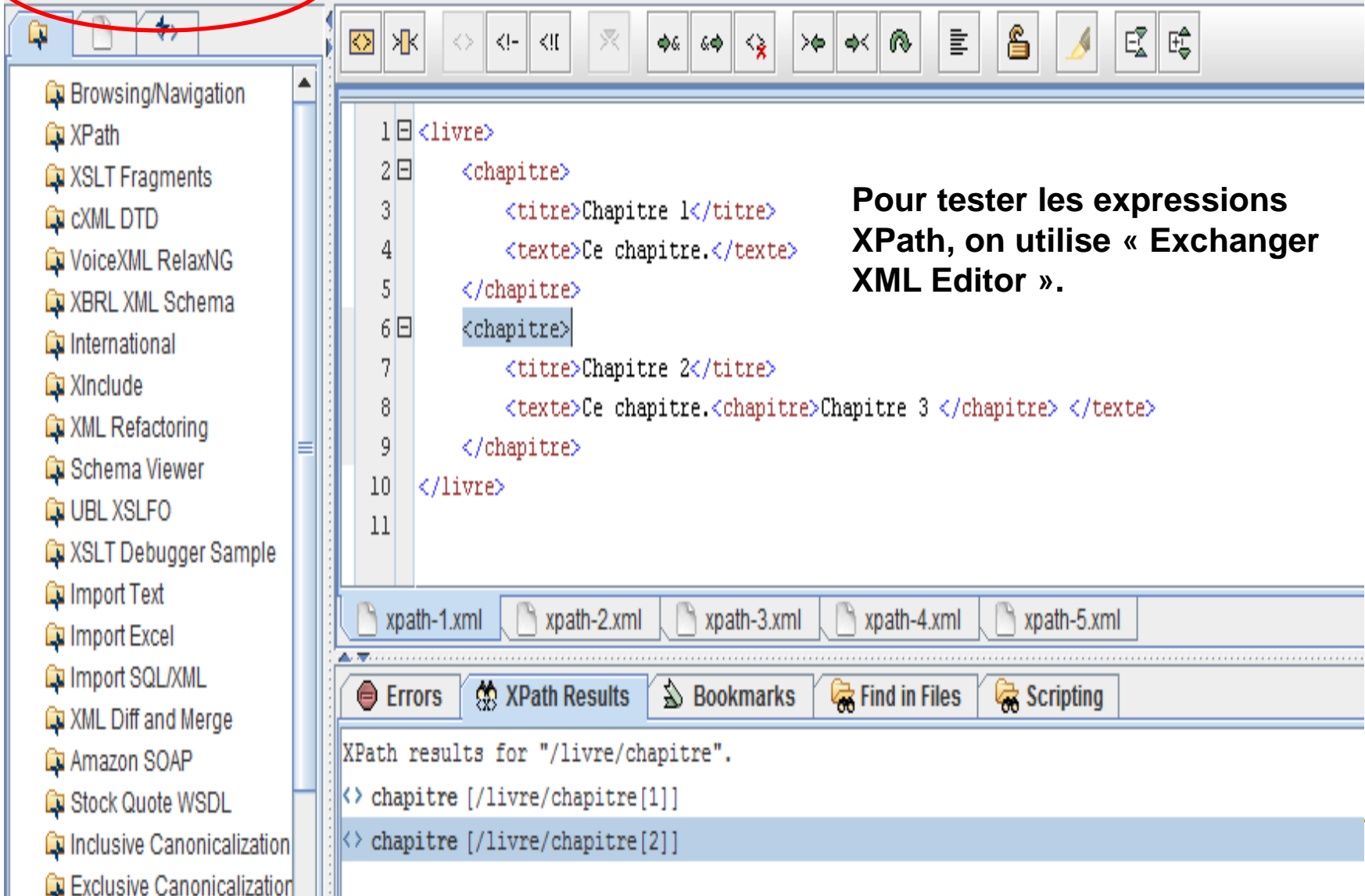
- Dans l'expression XPath « **/**livre/chapitre », la première barre oblique **/** (slash) indique que la recherche commence à partir du **nœud racine** (ou nœud de document/nœud document) de l'arbre du document XML. Il représente tout le document XML.
- Dans l'exemple ci-dessous, l'enfant direct du **nœud racine** **/** est le nœud d'élément racine (ou élément racine/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>
- **Tester avec « xpath-1.xml »**

Le nœud d'élément

XPath:  /livre/chapitre



The screenshot displays the Exchanger XML Editor interface. On the left, a sidebar lists various XML-related tools, with 'XPath' highlighted. 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, a toolbar contains icons for file operations and editing. A tab bar at the bottom shows several files: 'xpath-1.xml', 'xpath-2.xml', 'xpath-3.xml', 'xpath-4.xml', and 'xpath-5.xml'. The 'XPath Results' tab is active, displaying the results of the XPath expression `/livre/chapitre`:

XPath results for `"/livre/chapitre"`.

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

On the right side of the editor, a text box contains the following text:

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 de contexte).
- Exemple : **//**chapitre permet de localiser (sélectionner) tous les nœuds d'élément « chapitre » du document « xpath-1.xml ».
- <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>
- Tester avec « xpath-1.xml »

Le nœud d'élément

The screenshot shows an XML editor interface. At the top, the XPath input field is circled in red and contains the query `//*[local-name()='chapitre']`. The left sidebar lists various XML-related tools, with 'XPath' selected. The main editor 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 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 shows the 'XPath Results' for the query `//*[local-name()='chapitre']`, listing three results:

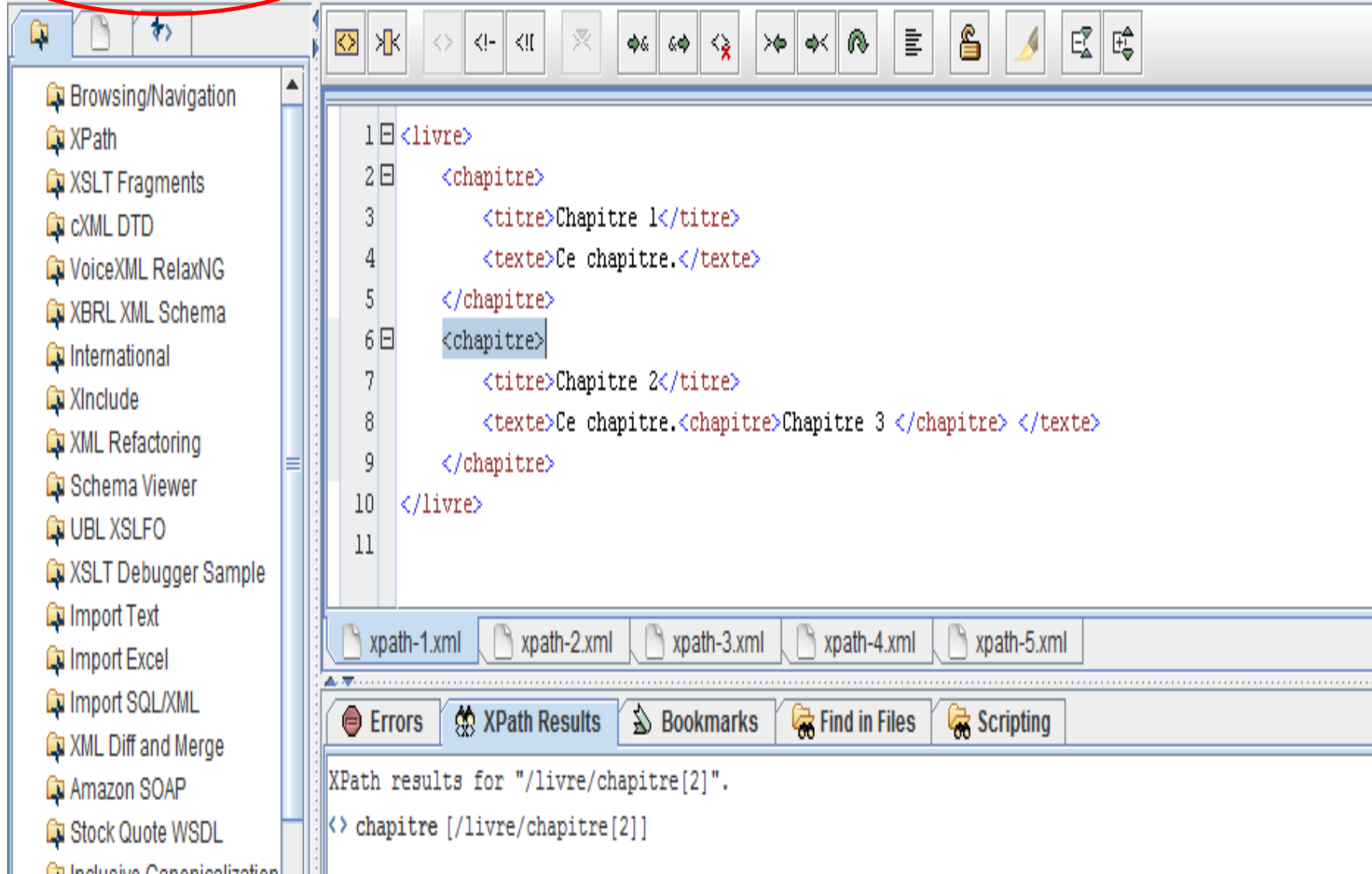
- `<chapitre [//livre/chapitre[1]]`
- `<chapitre [//livre/chapitre[2]]`
- `<chapitre [//livre/chapitre[2]/text/chapitre]`

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>
- Tester avec « 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 tools 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     <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 view 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 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**
- **//chapitre[2]/*/chapitre (utilisation du Wildcard * - voir plus loin slide 21)**
- **//chapitre[2]//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 the following items:

- 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

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

```
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 is a tab bar with five tabs: `xpath-1.xml`, `xpath-2.xml`, `xpath-3.xml`, `xpath-4.xml`, and `xpath-5.xml`.

At the bottom is a panel with five tabs: **Errors**, **XPath Results**, **Bookmarks**, **Find in Files**, and **Scripting**. The **XPath Results** tab is active, showing 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'élément possédant un nœud d'attribut, on utilise les crochets : **//personne[@prenom="Jack"]** (voir aussi les **Prédicats**, plus loin slide 40).
- Pour localiser tous les éléments « personne » possédant un nœud d'attribut « prenom » : **//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="01-05-1946"/>**
- **</liste>**
- **Tester avec « 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

- Pour localiser directement les nœuds d'attribut :
- **/nœud_courant/@type** (sélectionne/localise le nœud d'attribut type du nœud d'élément courant).
- Exemple :
- **/liste/personne/@nom**

Tester avec « xpath-2.xml »

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 XPath 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 nœud d'élément ou nœud d'attribut contenu dans le nœud d'élément courant. Cela correspond donc aux nœuds d'élément sans tenir compte de leur nom (voir aussi plus haut l'exemple du slide 12).
- Exemple 1:
- /AAA/BBB/* (localise tous les enfants directs du nœud d'élément courant /AAA/BBB).
- Exemple 2:
- /AAA/BBB/HHH/@* (localise tous les attributs (ou nœuds d'attribut) du nœud courant /AAA/BBB/HHH).
- Exemple 3:
- /AAA/BBB//* (localise tous les descendants du nœud d'élément courant /AAA/BBB en utilisant aussi la récursivité //).
- Tester avec « xpath-3.xml ».

Wildcard (joker) *

The screenshot shows an XML editor interface. The top menu bar includes File, Edit, View, Project, XML, Schema, Transform, Security, Tools, and Help. Below the menu is a toolbar with various icons. The XPath input field, located below the toolbar, contains the query `/AAA/BBB/*` and is circled in red. To the left of the main editor 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, Envelope XML Signature, Detached XML Signature, XQuery, and SVG. The main editor area displays 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 <FFF />
```

Below the XML document is a tab labeled `xpath-3.xml`. The bottom of the editor features a toolbar with buttons for Errors, XPath Results, Bookmarks, Find in Files, and Scripting. The XPath Results tab is active, showing the following results for the query `"/AAA/BBB/*"`:

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

At the bottom left, there is a section for Document Properties, which includes fields for Type, Encoding (UTF-8), Validation (Location defined in Do...), Schema (No location defined), and Completion (Inferred from Document).

Wildcard (joker) *

The screenshot shows an XML editor interface. The top menu bar includes File, Edit, View, Project, XML, Schema, Transform, Security, Tools, and Help. Below the menu is a toolbar with various icons. The XPath input field, located below the toolbar, contains the expression `/AAA/BBB/HHH/@*` and is circled in red. To the left of the main editor is a sidebar with a tree view of XML-related tasks, 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, Envelope XML Signature, Detached XML Signature, XQuery, and SVG. The main editor area displays 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 </BCC>
13 <BCC>
14   <BCD>
15     <HHH fromage="Cantal" />
16     <CDB />
17   </BCD>
18 </BCC>
19 <DDD>
20   <EEE />
```

At the bottom of the editor, there is a tab labeled `xpath-3.xml`. Below the tab 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 `"/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]
```

The Document Properties panel at the bottom left shows the document's encoding as UTF-8.

Wildcard (ioker) *

The screenshot displays the XMLSpy application interface. The 'XPath' tab is selected in the left-hand pane, and the XPath expression `/AAA/BBB/*` is entered in the text field. A red circle highlights this text field. The main editor shows an XML document structure with the following content:

```
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 />
```

Below the XML editor, the 'XPath Results' tab is active, showing the results of the query `/AAA/BBB/*`. The results are listed as follows:

```
<> 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]
```

The 'Document Properties' pane at the bottom left shows the following information:

- Type:
- Encoding: UTF-8
- Validation: Location defined in Doc...
- Schema: No location defined.
- Completion: Inferred from Document.

Le nœud d'attribut (opérateurs logiques/booléens)

- `//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="01-05-1946"/>`
- `</liste>`
- Tester avec « xpath-2.xml ».

Le nœud d'attribut (opérateurs logiques/booléens)

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

```
XPath results for "//personne[@prenom="Jack"] and //personne[@prenom="Nolan"]  
false
```

Le nœud d'attribut (opérateurs logiques/booléens)

Cette expression XPath génère un nœud d'élément :

- `//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="01-05-1946"/>`
- `</liste>`
- Tester avec « xpath-2.xml ».

The screenshot shows an XML editor with the following components:

- Menu Bar:** File, Edit, View, Project, XML, Schema, Transform, Security, Tools, Help.
- Toolbar:** Contains icons for file operations (open, save, print), editing (undo, redo, cut, copy, paste), and navigation (find, replace, etc.).
- XPath Input Field:** Displays the query `//personne[@prenom and @date_de_naissance]`.
- XML Document:** The main editor area shows an XML document with the following structure:

```
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="01-05-1946" />
6 </liste>
7
```
- File Explorer:** Located at the bottom, it shows a list of files: `xpath-1.xml`, `xpath-2.xml`, `xpath-3.xml`, `xpath-4.xml`, and `xpath-5.xml`.
- Toolbars:** Below the file explorer, there are tabs for **Errors**, **XPath Results**, **Bookmarks**, **Find in Files**, and **Scripting**.
- XPath Results Panel:** The **XPath Results** tab is active, showing the results of the query: `<> personne [/liste/personne[4]]`.

Que va sélectionner l'expression XPath suivante ?

`//personne[@prenom or @date_de_naissance]`

(voir la réponse dans le slide suivant)

```
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="01-05-1946" />
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 or @date_de_naissance]".

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


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

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

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


Le nœud d'attribut (opérateurs logiques/booléens)


- `//personne[not(@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="01-05-1946"/>`
- `</liste>`
- Tester avec « xpath-2.xml ».


XPath:  `//personne[not(@date_de_naissance)]`





```
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="01-05-1946" />
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(@date_de_naissance)]`".

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

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

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

Les fonctions

- `node()` → `//livre/chapitre/texte/node()`
- `position()` → `//personne[position()=3]`
- `last()` → `//personne [last()] == //personne[position()=last()]`
- `text()` → `//Capot/text()`
- `count()` → `count(//personne)`
- `name()` → `//*[name()='BBB']`
- `starts-with()` → `//*[starts-with(name(),'B')]`
- `string-length()` → `string-length(//Capot)`
- `string()` → `string(//chapitre)` traite uniquement le premier noeud trouvé
- `number()` → `number(//HHH/@att)` traite uniquement le premier noeud trouvé
- `contains()` → `contains(//Capot/text(), 'peu')` **renvoie un booléen**
- `concat()` → `concat(/Voiture/Carrosserie/Capot/text(),
'//Allumage/text())`
- `comment()` → `//comment()`

Les fonctions

The screenshot shows an XML editor with a document containing two chapters. The first chapter has a title 'Chapitre 1' and text 'Ce chapitre.'. The second chapter has a title 'Chapitre 2' and text 'Ce chapitre.<chapitre>Chapitre 3 </chapitre>'. The editor's interface includes a toolbar with buttons for 'Errors', 'XPath Results', 'Bookmarks', 'Find in Files', and 'Scripting'. The 'XPath Results' panel is active, displaying the results of the XPath query `"/livre/chapitre/texte/node()"`. The results are:

- `T Ce chapitre. [/livre/chapitre[1]/texte/text()]`
- `T Ce chapitre. [/livre/chapitre[2]/texte/text()]`
- `<> chapitre [/livre/chapitre[2]/texte/chapitre]`

A red arrow points to the XPath query in the results panel.

```
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
```

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 `"/livre/chapitre/texte/node()"`.


- `T Ce chapitre. [/livre/chapitre[1]/texte/text()]`
- `T Ce chapitre. [/livre/chapitre[2]/texte/text()]`
- `<> chapitre [/livre/chapitre[2]/texte/chapitre]`

Les fonctions

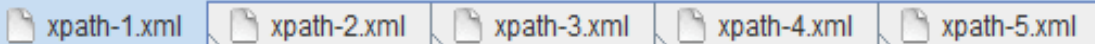
- Modifier l'expression XPath suivante pour extraire aussi le noeud de texte ***Chapitre 3*** ?
- `/livre/chapitre/texte/node()`
- Tester avec « xpath-1.xml ».






Les fonctions

XPath:  `/livre/chapitre/texte//node()` 







```
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
```



 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 `"/livre/chapitre/texte//node()"`.

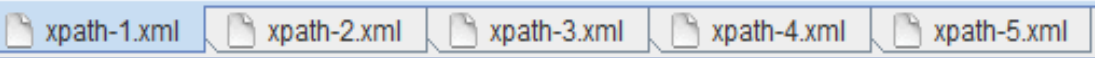
-  Ce chapitre. [/livre/chapitre[1]/texte/text()]
-  Ce chapitre. [/livre/chapitre[2]/texte/text()]
-  chapitre [/livre/chapitre[2]/texte/chapitre]
-  Chapitre 3 [/livre/chapitre[2]/texte/chapitre/text()]






Les fonctions

XPath:  `//texte//node()` 







```
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
```



 Errors  XPath Results  Bookmarks  Find in Files  Scripting


XPath results for "`//texte//node()`".


-  Ce chapitre. [`/livre/chapitre[1]/texte/text()`]
-  Ce chapitre. [`/livre/chapitre[2]/texte/text()`]
-  chapitre [`/livre/chapitre[2]/texte/chapitre`]
-  Chapitre 3 [`/livre/chapitre[2]/texte/chapitre/text()`]

Les fonctions

- Que va extraire l'expression XPath suivante ?
- `//personne[not(contains(@prenom, 'Billy'))]`
- Tester avec « xpath-2.xml ».

Les fonctions

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



```
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="01-05-1946" />
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

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 of various XML-related tools and schemas. The main area displays an XML document with the following structure:

```
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'. The 'XPath Results' tab is active, showing the results of the XPath query: "concat(/Voiture/Carrosserie/Capot/text(),' ',//Allumage/text())". The results are displayed as a single line of 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)
- **Tester avec « xpath-4.xml ».**

Les prédicats

XPath: `//atome[masse < 40]`

The screenshot shows an XML editor interface. On the left is a sidebar with a tree of icons for various XML-related tasks like Browsing/Navigation, XPath, XSLT Fragments, etc. The main 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 text are five tabs labeled `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 of the query `//atome[masse < 40]`:

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

Les prédicats

XPath: `//*[string-length(name ()) = 5]`

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" 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 toolbar with buttons for 'xpath-1.xml' through 'xpath-5.xml'. At the bottom, there is a tabbed interface with 'Errors', 'XPath Results', 'Bookmarks', 'Find in Files', and 'Scripting'. The 'XPath Results' tab is active, showing the results of the query `//*[string-length(name ()) = 5]`:

```
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]
```

At the bottom left, the 'Document Properties' panel shows 'Type:' and 'Encoding: iso-8859-1'.

Exercice

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