

6. XPath

6.1 Introduction

6.1.1 Fiche d'identité

Xpath est un langage **non-XML** pour identifier des sous-ensembles de documents **XML**.

XPath a fait l'objet de plusieurs recommandations du **W3C** :

📄 XPath 1.0 - novembre 1999

[\[http://www.w3.org/TR/xpath\]](http://www.w3.org/TR/xpath)

📄 XPath 2.0 - décembre 2010

[\[http://www.w3.org/TR/xpath20\]](http://www.w3.org/TR/xpath20)

📄 XPath 3.0 - avril 2014

[\[https://www.w3.org/TR/2014/REC-xpath-30-20140408/\]](https://www.w3.org/TR/2014/REC-xpath-30-20140408/)

ainsi que d'une **Candidate Recommendation** pour **XPath 3.1** datée de décembre 2015.

📄 Consulter le document le plus récent sur XPath 3.1

[\[https://www.w3.org/TR/xpath-31/\]](https://www.w3.org/TR/xpath-31/)

Xpath fait entre autres partie des "**accessoires**" de **XSLT**. C'est la raison pour laquelle les spécifications de **XPath** sur le site du **W3C** sont notamment accessibles depuis la page principale consacrée à **XSL** (*eXtensible Stylesheet Language*) sur le site du **W3C**.

📄 Consulter la page principale sur XSL

[\[http://www.w3.org/Style/XSL/\]](http://www.w3.org/Style/XSL/)

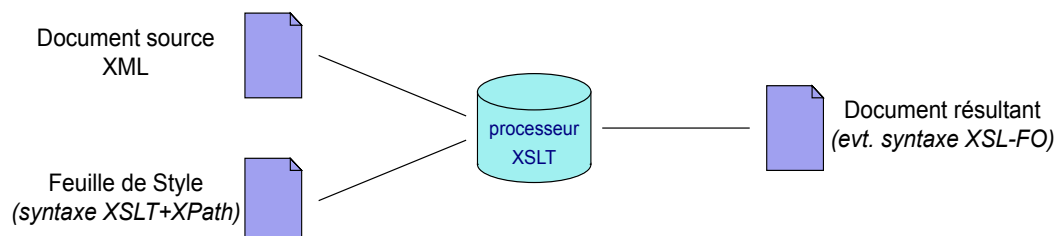
Comme **CSS** s'appuie sur des **sélecteurs** pour désigner certains éléments d'un document, **Xpath** est utilisé par **XSLT** pour sélectionner des éléments particuliers du document **XML** d'entrée et leur faire correspondre un modèle de traitement.

Toutefois, **Xpath** a été conçu comme un mécanisme générique. Il est également utilisé par **Xpointer** et **XQuery**.

6.1.2 Eléments de contexte

XPath est issu des besoins identifiés par le groupe de travail en charge de la définition de l'application de feuilles de styles **XSL** (*mi 1998*). Un an et quelques "**Working Drafts**" plus tard, la décomposition fonctionnelle de **XSL** était achevée.

On dispose depuis - de **XSLT** pour la transformation du document (*recommandation de novembre 1999*), - de **Xpath** (*utilisé par XSLT*) pour la localisation d'éléments dans un document (*novembre 1999*) - et de **XSL-FO** dont la recommandation a été finalisée plus tard, permettant la mise en forme d'un document, en vue notamment de l'impression (*octobre 2001*).



Remarquer que la feuille de style utilise une syntaxe propre à **XSLT**, qui s'appuie sur **XPath** pour désigner les éléments à "**mettre en page**" comme **CSS** s'appuie sur des sélecteurs.

Toutefois, **Xpath** ayant été conçu comme un langage générique, d'autres applications comme **XPointer** et **XQuery** bénéficient des fonctionnalités qu'il apporte.

6.1.3 Expressions XPath

Une expression **Xpath** peut renvoyer un noeud de l'arbre **XML** (*élément, attribut, ...*), un ensemble de noeuds (*node-set*), ou plus simplement une chaîne de caractères, un nombre ou un booléen :

`//étudiant[@id="E111"]` Renvoie un noeud

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/expressions-ex1.html>]

`//étudiant` Renvoie un ensemble de noeuds

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/expressions-ex2.html>]

`count(//étudiant)` Renvoie un nombre

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/expressions-ex3.html>]

`//étudiant[1]/nom = 'Deubaze'` Renvoie un booléen

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/expressions-ex4.html>]

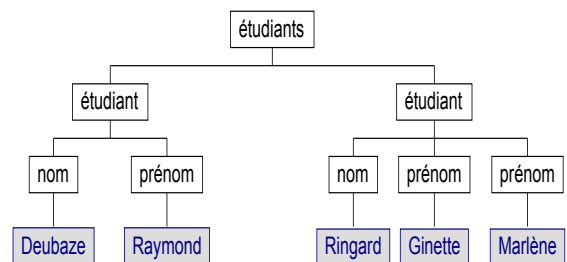
L'expression **XPath** toutefois la plus courante et la plus utile est le **chemin de localisation** (*location path* - cf. *exemples 1 et 2 ci-dessus*). Un chemin de localisation renvoie un noeud ou un ensemble de noeuds.

N.B. **XPath** travaille sur l'**arbre XML**. Certaines choses sont donc logiquement impossibles. Il n'est par exemple pas possible de savoir si du texte a été spécifié dans une section **CDATA** ou si un caractère particulier a été inséré avec un appel d'entité caractère.

6.1.4 Chemins de localisation

Reprenons le document **XML** bien connu à titre d'exemple :

```
<étudiants>
  <étudiant>
    <nom>Deubaze</nom>
    <prénom>Raymond</prénom>
  </étudiant>
  <étudiant>
    <nom>Ringard</nom>
    <prénom>Ginette</prénom>
    <prénom>Marlène</prénom>
  </étudiant>
</étudiants>
```



Un **chemin de localisation** désigne un élément particulier en navigant à travers l'arbre XML.

Syntaxiquement il ressemble au chemin d'accès d'une **URL** ou d'un fichier sur disque, permettant de naviguer à travers l'arbre des répertoires, à partir de la racine, pour atteindre un document particulier.

Un **chemin absolu** part de la racine du document. Au fur et à mesure de la progression vers l'élément recherché, le chemin de localisation se construit en séparant par des caractères **"/"** le nom des éléments traversés.

Voici par exemple le chemin de localisation permettant de désigner l'élément englobant :

`/étudiants`

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/locpath-ex1.html>]

6.1.5 Exemples de chemins de localisation

Si l'expression ci-dessous désigne bien l'élément englobant :

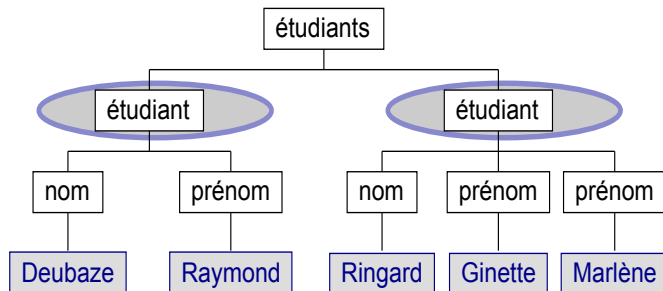
`/étudiants`

Que désigne l'expression suivante ?

`/étudiants/étudiant`

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/locpath-ex2.html>]

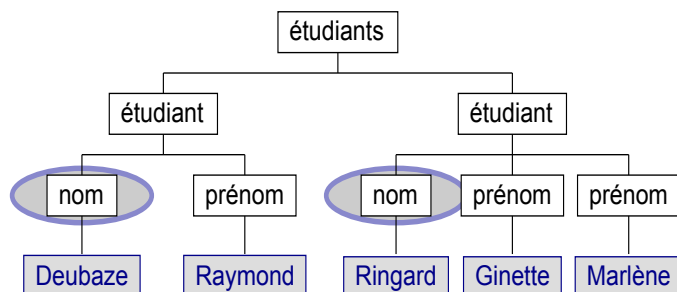


L'expression ci-dessous ?

`/étudiants/étudiant/nom`

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/locpath-ex3.html>]

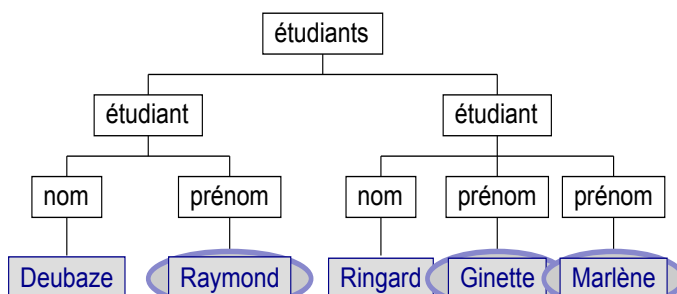


L'expression suivante fait appel à une fonction particulière :

`/étudiants/étudiant/prénom/text()`

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/locpath-ex4.html>]



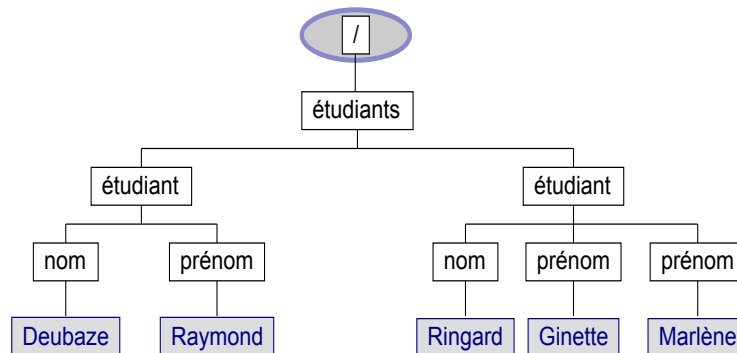
6.2 Chemins de localisation

6.2.1 Chemins de localisation absolus

Un **chemin de localisation absolu** permet de parcourir l'arbre **XML** à partir de la racine. L'expression qui représente un tel chemin commence toujours par le caractère **"/"**.

N.B. L'élément racine (*root*) est un élément **XML** abstrait. Il est représenté par le chemin :

/



En circulant le long d'un chemin de localisation le nom qui suit un caractère **"/"** désigne l'ensemble des éléments qui portent ce nom, descendants directs (*enfants*) de l'élément précédent du chemin.

/étudiants/étudiant/nom

6.2.2 Sélection des descendants

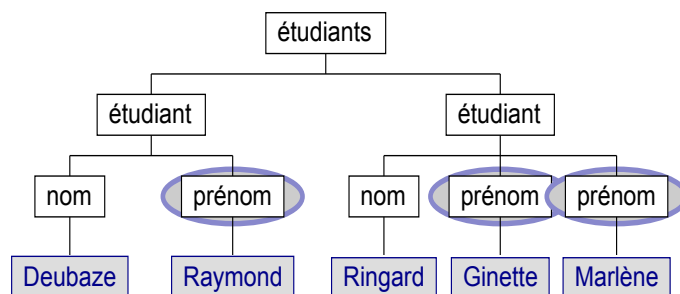
Il est possible de sélectionner un élément parmi l'**ensemble des descendants** à une certaine étape du chemin (*et non pas uniquement parmi les enfants directs*), en faisant précéder le nom de cet élément par un double **"/"**.

//prénom

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/abbrdesc-ex1.html>]

sélectionne l'ensemble des éléments **"prénom"** du document.



Cette notation peut apparaître à n'importe quelle étape du chemin :

/étudiants//prénom

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/abbrdesc-ex2.html>]

sélectionne l'ensemble des éléments **"prénom"**, descendants de l'élément **"étudiants"** (*enfants directs ou descendants lointains*).

6.2.3 Chemins de localisation relatifs

XPath permet également de travailler dans le contexte d'un élément particulier.

Les chemins de localisation partant de l'élément courant sont appelés des chemins **relatifs** et ne comportent pas de caractère "/" initial.

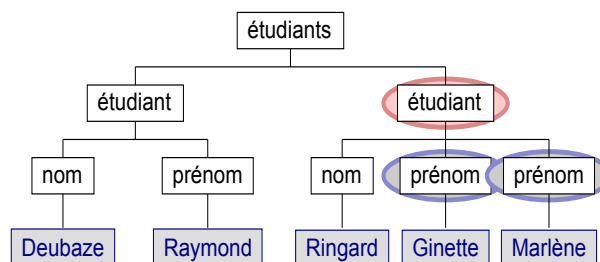
Supposons que l'élément courant soit le deuxième "*étudiant*" du document :

Que représente le chemin suivant ?

`prénom`

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/relpath-ex1.html>]

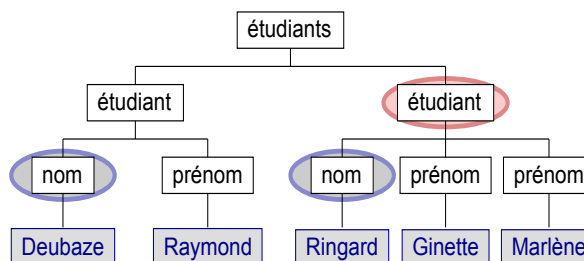


Et celui-ci ?

`../nom`

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/relpath-ex2.html>]



6.2.4 Sélection des attributs

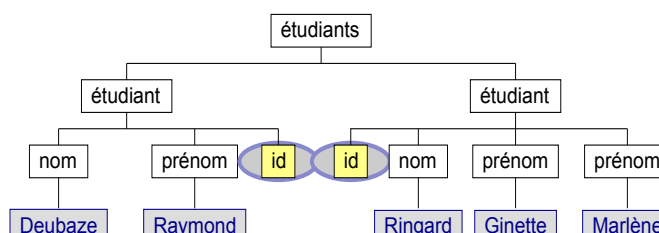
Dans l'arbre **XML**, les **attributs** sont des enfants du noeud correspondant à leur élément parent.

La syntaxe **XPath** pour sélectionner un attribut, consiste à préfixer son nom à l'aide du caractère "@" :

`//étudiant/@id`

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/abbrattr-ex1.html>]



6.3 Sélecteurs, axes et prédicats

6.3.1 Sélecteurs de noeuds

Pour désigner des noeuds dans l'arbre, **XPath** utilise des **sélecteurs de noeuds** (*node tests*).

Les seuls sélecteurs de noeuds utilisés jusqu'ici ont été des noms d'éléments. Toutefois, il y a bien d'autres façons de sélectionner des noeuds.

> élément

C'est le sélecteur vu jusqu'ici. Il permet de désigner les éléments de nom "*élément*".

```
//prénom
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex1.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex1.html)

> *

Il s'agit d'un joker qui désigne n'importe quel **élément**.

```
//étudiant[2]/*
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex2.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex2.html)

> text()

Cette expression a déjà été rencontrée, et permet de sélectionner les noeuds texte.

```
//étudiant[2]/prénom/text()
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex25.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex25.html)

> comment()

Cette fonction permet de sélectionner les commentaires.

```
//étudiant[2]/comment()
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex3.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex3.html)

> processing-instruction()

Désigne les instructions de traitement.

```
//étudiant[2]/processing-instruction()
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex4.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex4.html)

> processing-instruction('proc')

Désigne les instructions de traitement à destination du processeur "*proc*".

```
//étudiant[2]/processing-instruction('test1')
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex5.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex5.html)

> node()

Sélectionne tous les noeuds, quel que soit leur type.

```
//étudiant[2]/node()
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex6.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/node-test-ex6.html)

6.3.2 Prédicats

Les chemins de localisation vus jusqu'à présent désignaient toujours une classe de noeuds simplement désignés par leur nom. Les **prédicats** permettent de sélectionner les éléments qui répondent à une condition donnée, à une étape quelconque du chemin de localisation.

Tentons d'extraire du document ci-dessous le prénom usuel du deuxième étudiant :

```
<étudiants>
  <étudiant id="E111">
    <nom>Deubaze</nom>
    <prénom>Raymond</prénom>
  </étudiant>
  <étudiant id="E314">
    <nom>Ringard</nom>
    <prénom type="usuel">Ginette</prénom>
    <prénom>Marlène</prénom>
  </étudiant>
</étudiants>
```

L'expression suivante désigne l'ensemble de tous les prénoms usuels :

```
//prénom[@type='usuel']
```

Exemple :

<http://dmolinarius.github.io/demofiles/mod-84/xpath/predicates-ex1.html>

Toutefois, cela ne répond que partiellement à la question. Le nième élément peut se désigner à l'aide d'une condition réduite à un nombre :

```
//étudiant[2]
```

La valeur du prénom usuel du deuxième étudiant se trouve donc à l'aide de l'expression :

```
//étudiant[2]/prénom[@type='usuel']/text()
```

Exemple :

<http://dmolinarius.github.io/demofiles/mod-84/xpath/predicates-ex2.html>

6.3.3 Exemples de prédicats

L'expression qui décrit la condition de sélection des noeuds (*prédicat*) peut être arbitrairement complexe et faire appel à des fonctions **XPath** prédéfinies (*vues plus loin*).

De fait le prédicat réduit à un nombre, qui permet d'extraire le nième noeud d'un ensemble de noeuds, est une version simplifiée de l'expression :

```
element[position()=n] // équivalent à element[n]
```

Comment peut-on sélectionner à l'aide d'un chemin de localisation absolu le prénom de l'étudiant dont le nom est *"Deubaze"* ?

Solution :

<http://dmolinarius.github.io/demofiles/mod-84/xpath/predicates-ex3.html>

Comment sélectionner le nom de tous les étudiants ayant plus d'un prénom ?

Solution :

<http://dmolinarius.github.io/demofiles/mod-84/xpath/predicates-ex4.html>

Le prénom du dernier étudiant ?

Solution :

<http://dmolinarius.github.io/demofiles/mod-84/xpath/predicates-ex5.html>

6.3.4 Etapes de localisation non abrégées

Les chemins de localisation vus jusqu'à présent spécifiaient chacune des étapes en notation abrégée.

Voici la notation abrégée, puis non abrégée de quelques chemins de localisation simples :

```
/étudiants/étudiant/nom
/child::étudiants/child::étudiant/child::nom
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/fullstep-ex1.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/fullstep-ex1.html)

```
../étudiant[1]/prénom
parent::node()/child::étudiant[1]/child::prénom
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/fullstep-ex2.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/fullstep-ex2.html)

```
prénom/@type
child::prénom/attribute::type
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/fullstep-ex3.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/fullstep-ex3.html)

```
../étudiant[@id='E111']/prénom
parent::node()/child::étudiant[attribute::id='E11']/child::prénom
```

Exemple :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/fullstep-ex4.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/fullstep-ex4.html)

6.3.5 Axes

Le terme qui apparaît devant le caractère "." s'appelle un **axe**. Il sert à indiquer la direction dans laquelle il faut progresser pour l'étape suivante du chemin de localisation.

XPath propose **treize axes** différents dont certains possèdent une notation abrégée, d'autres non.

> Axes possédant une notation abrégée

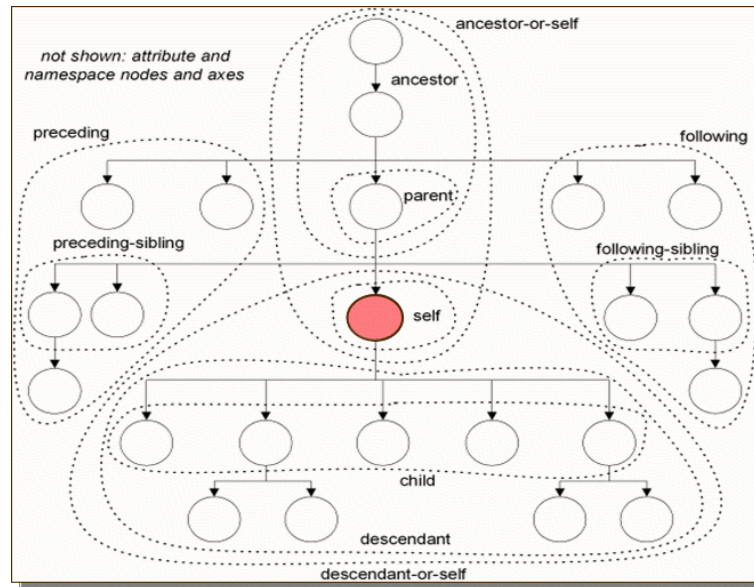
Axe	Ensemble concerné	Notation abrégée
child::	enfants directs du noeud courant	(axe par défaut)
self::	noeud courant	.
parent::	parent du noeud courant	..
attribute::	attributs du noeud courant	@
descendant-or-self::	descendants, noeud courant compris	//

> Autres axes

Axe	Ensemble concerné	Sens de parcours
ancestor::	ascendants du noeud courant	inverse
ancestor-or-self::	ascendants, noeud courant compris	inverse
descendant::	descendants du noeud courant	direct
following::	suivants dans l'ordre du document	direct
following-sibling::	frères suivants, dans l'ordre du doc.	direct
preceding::	précédents dans l'ordre du document	inverse
preceding-sibling::	frères précédents, dans l'ordre du doc.	inverse
namespace::	espaces de noms	-

6.3.6 Récapitulatif des axes

La figure suivante récapitule l'ensemble des axes :



Source : Crane Softwrights

6.3.7 Syntaxe généralisée des étapes de localisation

La syntaxe généralisée des étapes de chemins de localisation comprend un **axe**, un **sélecteur de noeuds** (*node test*) et un prédicat :

axe::sélecteur[prédicat]

Cette syntaxe concerne **chacune des étapes**.

Considérons à titre d'exemple, un document dont l'élément principal s'appelle *"option"* et comporte une liste de personnes :

```
<personnes>
  <personne id="P1" civilité="M.">
    <prénom>Jérôme</prénom>
    <nom>Gourdin</nom>
    <phone>06 54 32 10 98</phone>
    <email>J-Go@freedot.com</email>
    <adresse>6, impasse des fées</adresse>
    <ville>Maubeuge</ville>
  </personne>
  . . .
</personnes>
```

suivie par une liste de cours :

```
<cours intitulé="Technologies de validation" prof="P1">
  <étudiant id="P4">
    <note>12.5</note>
    <absences>1</absences>
    <comment>Devrait se mettre à jour</comment>
  </étudiant>
</cours>
. . .
```

- Comment trouver le prénom de la personne qui porte le nom "Deubaze" ?

Réponse :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/syntax-ex1-option.html>]

- En supposant les personnes classées par ordre alphabétique du nom, quel est le nom de l'avant dernière personne ?

Réponse :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/syntax-ex3-option.html>]

- Quel est l'intitulé des cours auxquels est inscrit "Antoine Nette" ?

Réponse :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/syntax-ex4-option.html>]

- Comment obtenir la liste des noms des inscrits au premier cours ?

Réponse :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/syntax-ex5-option.html>]

6.3.8 Chemins composés

Il est possible de désigner une liste d'objets composée à l'aide de plusieurs chemins de localisation. On parle alors de **chemin composé** (*compound location*).

La syntaxe générale est :

`chemin_1 | chemin_2`

N.B. Les exemples ci-dessous s'appuient sur le document vu précédemment.

- Comment trouver le nom et le prénom de la personne qui porte l'id "P1" ?

Réponse :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/compound-ex2-option.html>]

- On désirerait l'intitulé du second cours, ainsi que le nom et le prénom du professeur.

Réponse :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/compound-ex3-option.html>]

N.B. L'ordre des objets renvoyés par une expression **XPath** est quelconque, non significatif, et notamment indépendant de l'ordre dans lequel sont arrangées les composantes d'une requête composée.

6.4 Expressions générales

6.4.1 Opérateurs

Outre les chemins de localisation qui renvoient un ensemble de noeuds (*node-set*), **Xpath** permet de spécifier des expressions qui renvoient une valeur du type numérique, booléenne ou chaîne de caractères.

Pour les opérations numériques, **Xpath** possède 5 opérateurs arithmétiques :

Op.	Opération	Exemple
+	addition	<code>1 + count(//cours[1]/étudiant)</code>
-	soustraction	<code>count(//personne) - count(//cours[1]/étudiant)</code>
*	multiplication	<code>0.5 * //étudiant[@id='P3']/note</code>
div	division	<code>sum(//cours[1]/étudiant/note) div count(//cours[1]/étudiant)</code>
mod	modulo	<code>count(//cours[1]/étudiant) mod 2</code>

Exemples :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex1-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex1-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex2-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex2-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex3-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex3-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex4-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex4-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex5-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex5-option.html)

Les opérateurs de comparaison renvoient un résultat booléen (*utile pour construire des prédicats*) :

Op.	Test	Exemple
=	égalité	<code>//personne[@id = 'P1']</code>
!=	non-égalité	<code>//cours[@prof != 'P1']/@intitulé</code>
<	inférieur	<code>//personne[@id=//étudiant[note < 10]/@id]</code>
>	supérieur	<code>//personne[@id=//étudiant[absences > 2]/@id]/nom</code>
<=	inférieur ou égal	<code>//personne[@id=//étudiant[absences <= 1]/@id]/nom</code>
>=	supérieur ou égal	<code>//personne[@id=//étudiant[absences >= 2]/@id]/nom</code>

Exemples :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex6-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex6-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex7-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex7-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex8-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex8-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex9-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex9-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex10-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex10-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex11-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex11-option.html)

Les opérateurs logiques sont classiques :

Op.	Opérateur	Exemple
and	ET logique	<code>//personne[@id=//étudiant[absences>=2 and note<11]/@id]/nom</code>
or	OU logique	<code>//personne[@id=//étudiant[absences>2 or note<10]/@id]/nom</code>

Exemples :

[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex12-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex12-option.html)
[\[http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex13-option.html\]](http://dmolinarius.github.io/demofiles/mod-84/xpath/operators-ex13-option.html)

6.4.2 Fonctions XPath

Xpath possède également un certain nombre de **fonctions**, utilisables notamment dans les prédicats.

Certaines des fonctions **XPath** (comme *position()* ou *last()*) ont déjà été rencontrées aux détours des exemples précédents :

🔧 **last()** :

[cf. paragraphe 6.5.1.1]

renvoie l'index du dernier élément dans le contexte courant.

🔧 **position()** :

[cf. paragraphe 6.5.1.2]

renvoie l'index de l'élément courant dans le contexte courant.

🔧 **count(node-set)** :

[cf. paragraphe 6.5.1.3]

renvoie le nombre de noeuds de l'ensemble spécifié.

N.B. Les fonctions *last()* et *position()* sont souvent utilisées au sein de prédicats, bien que la seconde puisse également servir par exemple à générer une numérotation automatique.

6.5 Fonctions

6.5.1 Fonctions courantes

6.5.1.1 La fonction last()

> last()

Renvoie l'index du dernier élément dans le contexte courant.

```
last()
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/last-ex1.html>]

N.B. Cette fonction est en général utilisée pour la construction de prédicats comme celui-ci :

```
//personne[last()]
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/last-ex2.html>]

qui est la version simplifiée de :

```
//personne[position()=last()]
```

6.5.1.2 La fonction position()

> position()

Renvoie l'index de l'élément courant dans le contexte courant.

```
position()
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/position-ex1.html>]

N.B. Cette fonction est en général utilisée pour la construction de prédicats (*voir aussi la notation abrégée*) :

```
//personne[position()=3] ou plus simplement //personne[3]
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/position-ex2.html>]

6.5.1.3 La fonction count()

> count(node-set)

Renvoie le nombre de noeuds de l'ensemble spécifié.

```
count(//personne)
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/count-ex1.html>]

N.B. Cette fonction est très utile pour numéroter automatiquement les têtes de chapitre, sections, et autres paragraphes : il suffit de compter le nombre d'éléments identiques à l'élément courant qui le précède au sein du document...

```
count(preceding-sibling::personne)
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/count-ex2.html>]

6.5.1.4 La fonction lang()

> lang(str)

Renvoie *"true"* si la langue du noeud courant (cf. *xml:lang*) est celle spécifiée par la chaîne passée.

```
<text xml:lang="en">
  This is an example of english text.
</text>
```

```
//text[lang('en')]
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/lang-ex1.html>]

L'avantage de cette fonction par rapport à un test visant à déterminer la valeur de l'attribut *"xml:lang"* est que la langue est héritée depuis les parents vers les enfants, même si ceux-ci ne spécifient pas cet attribut :

```
<exemple xml:lang="fr">
  <text>
    Voici un exemple de texte en français.
  </text>
</exemple>
```

```
//text[lang('fr')]
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/lang-ex2.html>]

Un autre avantage de cette fonction est que le standard des formats de langue permet de spécifier des variantes locales comme *"fr-FR"*, *"fr-BE"* ou *"fr-CA"*. Si une telle variante est présente, alors une requête vers la langue générique retiendra la variante :

```
<text xml:lang="en-UK">
  This is an example of english text.
</text>
```

```
//text[lang('en')]
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/lang-ex4.html>]

6.5.2 Fonctions renvoyant des noms

6.5.2.1 La fonction local-name()

> local-name(node-set)

Renvoie le nom local (i.e. sans le préfixe d'espace de noms) du noeud spécifié ou du premier noeud de l'ensemble.

```
local-name (//*[@id='P1'])
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/local-name-ex1.html>]

N.B. Pour obtenir le nom d'un élément avec son préfixe, voir la fonction *name()*...

6.5.2.2 La fonction name()

> name(node-set)

Renvoie le nom complet (*i.e. avec le préfixe d'espace de noms*) du noeud spécifié ou du premier noeud de l'ensemble.

```
name (//*[ @id='P1' ] )
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/name-ex1.html>]

N.B. Pour obtenir le nom d'un élément **sans** son préfixe, voir la fonction *local-name()*...

6.5.2.3 La fonction namespace-uri()

> namespace-uri(node-set)

Renvoie l'URI d'espace de noms du noeud spécifié ou du premier noeud de l'ensemble.

```
namespace-uri (//*[ @id='P1' ] )
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/namespace-uri-ex1.html>]

6.5.3 Fonctions de chaînes

6.5.3.1 La fonction string()

> string(object)

Renvoie la valeur de l'objet sous forme de chaîne de caractères.

```
string (//*[ @id='P1' ] )
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/string-ex1.html>]

N.B. Cette fonction s'applique à toutes sortes d'arguments : élément, noeud texte, attribut, ou ensemble de noeuds (*node-set*). Elle se contente de renvoyer le contenu textuel de l'ensemble des noeuds passés en argument.

Voici un exemple avec un attribut :

```
string (//*[ @id='P6' ]/civilité)
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/string-ex2.html>]

6.5.3.2 La fonction string-length()

> string-length(string)

Renvoie le nombre de caractères de la chaîne "*string*".

```
string-length (//cours[1]/@intitulé)
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/string-length-ex1.html>]

6.5.3.3 La fonction substring()

> substring(str, start, [len])

Renvoie la sous-chaîne de "str" commençant à la position "start," comportant "len" caractères ou jusqu'à la fin de "str".

```
substring(phone, 1, 2)
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/substring-ex1.html>]

N.B. Le dernier argument est optionnel. En son absence la sous-chaîne renvoyée comprend tous les caractères jusqu'à la fin de la chaîne source :

```
substring(@id, 2)
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/substring-ex2.html>]

6.5.3.4 La fonction substring-before()

> substring-before(str1, str2)

Renvoie la sous-chaîne de "str1" située avant la première occurrence de "str2", ou la chaîne vide si "str2" n'est pas contenue dans "str1".

```
substring-before(email, '@')
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/substring-before-ex1.html>]

6.5.3.5 La fonction substring-after()

> substring-after(str1, str2)

Renvoie la sous-chaîne de "str1" située après la première occurrence de "str2", ou la chaîne vide si "str2" n'est pas contenue dans "str1".

```
substring-after(email, '@')
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/substring-after-ex1.html>]

6.5.3.6 La fonction concat()

> concat(str1, str2, ...)

Renvoie la chaîne obtenue en concaténant les arguments passés.

```
concat(substring(prénom, 1, 1), substring(nom, 1, 1), substring(nom, string-length(nom)))
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/concat-ex1.html>]

N.B. L'exemple ci-dessus renvoie le fameux trigramme cher à certains, avec le défaut que la dernière lettre se trouve être en mibuscules ...

6.5.3.7 La fonction `normalize-space()`

> `normalize-space(str)`

Renvoie "*str*" compactée *i.e.* sans espace initial ou final et les espaces internes remplacés par un espace unique.

```
normalize-space (//exemple)
```

Exemple non normalisé :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/normalize-space-ex1.html>]

Exemple normalisé :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/normalize-space-ex2.html>]

6.5.3.8 La fonction `translate()`

> `translate(str1, str2, str3)`

Renvoie "*str1*" dont les caractères présents dans "*str2*" ont été remplacés par ceux de "*str3*".

```
concat (
  substring (prénom, 1, 1) ,
  substring (nom, 1, 1) ,
  translate (
    substring (nom, string-length (nom) ) ,
    'abcdefghijklmnopqrstuvwxyz' ,
    'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  )
)
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/translate-ex1.html>]

N.B. L'exemple ci-dessus renvoie le fameux trigramme cher à certains, en prenant soin de mettre la dernière lettre en majuscules ...

6.5.3.9 La fonction `starts-with()`

> `starts-with(str1, str2)`

Renvoie "*true*" si "*str1*" commence par "*str2*".

```
//personne[starts-with(nom, 'D')] /prénom
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/starts-with-ex1.html>]

N.B. Cette fonction est plus particulièrement employée dans le contexte de prédicats.

6.5.3.10 La fonction `contains()`

> `contains(str1, str2)`

Renvoie "*true*" si "*str1*" contient "*str2*".

```
//personne/adresse[contains(., 'place')]
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/contains-ex1.html>]

N.B. Cette fonction est plus particulièrement employée dans le contexte de prédicats.


6.5.4 Fonctions numériques

6.5.4.1 Fonctions XPath

> Fonctions numériques

number(object) :

renvoie la valeur numérique de l'objet (*conversion*).

 *sum(node-set)* :

[cf. paragraphe 6.5.4.2]

renvoie la somme des valeurs numériques de l'ensemble des noeuds passés en argument.

floor(number) :

renvoie le plus grand entier inférieur au nombre passé.

ceiling(number) :

renvoie le plus petit entier supérieur au nombre passé.

round(number) :

renvoie l'entier le plus proche du nombre passé.

6.5.4.2 La fonction sum()

> sum(node-set)

Renvoie la somme des valeurs numériques de l'ensemble des noeuds passés en argument.

```
sum(//cours[1]//note) div count(//cours[1]//note)
```

Exemple :

[<http://dmolinarius.github.io/demofiles/mod-84/xpath/functions/sum-ex1.html>]

6.5.5 Fonctions booléennes

6.5.5.1 Fonctions XPath

> Fonctions booléennes

boolean(object) :

renvoie la valeur booléenne de l'objet (*conversion*).

not(boolean) :

renvoie la négation de l'argument passé.

true() :

renvoie *"true"*.

false() :

renvoie *"false"*.