

TIRÈME SARL

# XPath, XML Path Language

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [XML Query](#)

[XPath](#) est un langage qui permet d'adresser, de désigner, des objets structurels contenus dans un document [XML](#). Il est conçu pour être utilisé, tant par [XSLT](#) que par [XPointer](#) ou, encore, par [XML Query](#), qui ajouteront à cette recommandation partagée les fonctionnalités propres qui leurs sont nécessaires. [XPath](#) est donc une recommandation de base, externalisée pour être partagée par un ensemble de recommandations de plus haut niveau.

## Objectifs

---

Si un document [XML](#) contient en même temps des données et des informations permettant d'identifier la structure et le sens de ces données, il est alors utile de pouvoir s'appuyer sur cette information pour désigner une partie d'un document [XML](#).

C'est utile lorsque l'on réalise des applications de présentation, par exemple, pour faire une table des matières où l'on ne veut sélectionner que des titres. C'est également utile lorsque l'on veut réaliser des hyperliens sur des documents que l'on ne peut pas modifier pour leur ajouter des ancrs et que l'on souhaite pourtant désigner (par exemple, le deuxième alinéa du troisième chapitre d'un document).

Du coup et pour tous ces propos, il est nécessaire d'avoir un langage de désignation d'objets dans un document ; c'est l'objectif de [XPath](#). Du point de vue du [W3C](#), l'objectif de [XPath](#) est aussi de devenir un *standard de base*, réutilisable dans des recommandations de plus haut niveau.

## Principes

---

Pour désigner un objet dans un document, [XPath](#) propose, d'une part, un langage d'adressage d'objets et, d'autre part, un ensemble de fonctions permettant d'augmenter le pouvoir d'expression du langage.

Le langage d'adressage utilise des chemins, pour désigner un ensemble d'objets ; la désignation de chemins se repose sur des notions d'axes et de sélection.

Plus précisément, un chemin, qui peut être absolu ou relatif, utilise des éléments (la recommandation parle d'*étapes*) de localisation qui se décomposent en :

un axe, choisi parmi les attributs (*attribute*), les fils (*child*), les parents (*parent*) ou tous les ancêtres (*ancestor*), les frères (*following-sibling* ou *preceding-sibling*), etc. ;

un test, sur un type de noeud, par exemple, les ancêtres de type `chapter` (`ancestor::chapter`);  
 un ou plusieurs prédicats, par exemple, le dernier des fils de type `chapter` (`child::chapter[position()=Last()]`).

Ces étapes se composent entre elles en utilisant l'opérateur `/`. Par exemple, le titre du parent de type `chapitre` s'écrira `ancestor::chapter/titre`.

Le résultat d'un chemin d'adressage est un *ensemble de noeuds (node set)* dans l'arbre XML du document, noeuds qui sont traités ensuite par des processus applicatifs *ad hoc*, qui intègrent XPath comme chemin d'adressage.

La syntaxe complète de XPath étant, par moment, trop "verbeuse", la recommandation inclut une syntaxe abrégée, allant directement à l'essentiel. Ainsi, `//p` sera une abréviation de `self::node()/child::p`: tous les éléments `p` contenus dans le sous-arbre en cours d'exploration (utilisation de l'axe `self`) et, cela, quelle que soit leur position dans ce sous-arbre.

Enfin, pour définir les prédicats, XPath propose un langage d'expressions, utilisant un ensemble de fonctions de base, telles les extractions de chaînes de caractères (`concat`, `contains`, `start-with`, ...), les explorations de position d'un noeud parmi ses frères (`position`, `last`, `count`, ...), etc.

## Modèle de données XPath et Infoset

---

Le modèle de données d'XPath comporte 7 types de noeuds (*Infoset* parle d'*éléments d'information*): racine, élément, texte, attribut, espace de nom, *processing instruction* et commentaire.

Les deux différences majeures avec la recommandation *Infoset* sont, d'une part, que les attributs sont considérés comme étant des noeuds d'arbre (même s'ils ont un statut spécifique, en ce sens qu'ils n'ont pas de fils) et, d'autre part, que les noeuds "texte", pour XPath, contiennent un ensemble de caractères, alors qu'*Infoset* distingue chaque caractère individuellement. La recommandation XPath définit alors, en annexe, comment interpréter les noeuds en fonction d'XPath.

## Recommandations(s)

---

### Langage XML Path

Recommandation, version 1.0, du 16-11-1999  
 Document sur <http://xmlfr.org/w3c/TR/xpath>

### XML Path Language

Recommandation, version 2.0, du 23-01-2007  
 Document sur <http://www.w3.org/TR/xpath20/>

### XPath Requirements

Projet en cours, version 2.0, du 03-06-2005  
 Document sur <http://www.w3.org/TR/xpath20req>

### XQuery 1.0 and XPath 2.0 Functions and Operators

Recommandation, version 1.0, du 23-01-2007  
 Document sur <http://www.w3.org/TR/xpath-functions/>

### XQuery 1.0 and XPath 2.0 Data Model (XDM)

Recommandation, version 20070123, du 23-01-2007  
 Document sur <http://www.w3.org/TR/xpath-datamodel/>

 *XQuery and XPath Full Text 1.0*

Recommandation Candidate, version 1.0, du 28-01-2010

Document sur <http://www.w3.org/TR/xpath-full-text-10/>

 *XQuery and XPath Full Text 1.0 Requirements*

Projet en cours, version 20080516, du 16-05-2008

Document sur <http://www.w3.org/TR/xpath-full-text-10-requirements/>

 *XQuery and XPath Full Text 1.0 Use Cases*

Projet en cours, version 1.0, du 28-01-2010

Document sur <http://www.w3.org/TR/xmlquery-full-text-use-cases>