

Les chemins XPath canoniques

par [Mathieu Lemoine](#)

Date de publication :

Dernière mise à jour : 09/10/2006

La norme XPath permet un certains nombre de notations courtes permettant d'abrégé les chemins XPath, nous essaieront de voir à quoi correspondent ces diverses notations courtes.

Introduction

I - Les axes abrégés

- I-1 - `//*` : l'axe descendant
- I-2 - `/*` : l'omission de l'axe (l'axe child).
- I-3 - `@*` : l'axe attribute
- I-4 - `..` : l'axe parent
- I-5 - `.` : l'axe self

II - Les prédicats courts

- II-1 - Les prédicat numérique
- II-2 - Les prédicats de type noeud
- II-3 - Les prédicats implicites

Introduction

La norme XPath permet un certains nombre de notations courtes permettant d'abrégé les chemins XPath.

Il y a deux catégories d'abréviations : celles portant sur les axes et celles portant sur les prédicats. Les abréviations portant sur les prédicats peuvent elles-même être regroupées en deux groupes distincts :

- Celles portant sur le prédicat en entier [les prédicats courts].
- Celles portant sur des sous-expressions à l'intérieur du prédicat.

Dans tous les cas, ces abréviations interviennent en fait implicitement au niveau des collisions de types.

Pour plus d'informations sur les types XPath, voir le tuto d'[Erwy](#) : [types en Xpath 1.0](#)

I - Les axes abrégés

I-1 - //* : l'axe descendant

Le double slash, qui est utilisé très fréquemment correspond à l'application de l'axe descendant.

AAA/* ou AAA/descendant::*

```
<root>
  <AAA>
    <BBB>
      <CCC/>
    </BBB>
  <BBB>
    <CCC/>
  </BBB>
</AAA>
<AAA>
  <BBB/>
</AAA>
<AAA>
  <CCC/>
</AAA>
</root>
```

//CCC ou /descendant::CCC

```
<root>
  <AAA>
    <BBB>
      <CCC/>
    </BBB>
  <BBB>
    <CCC/>
  </BBB>
</AAA>
<AAA>
  <BBB/>
</AAA>
<AAA>
  <CCC/>
</AAA>
</root>
```

I-2 - /* : l'omission de l'axe (l'axe child).

Quand on n'indique pas d'axe, c'est l'axe child qui est utilisé par défaut.

AAA/* ou AAA/child::*

```
<root>
  <AAA>
    <BBB>
      <CCC/>
    </BBB>
  <BBB>
    <CCC/>
  </BBB>
</AAA>
<AAA>
  <BBB/>
</AAA>
<AAA>
  <CCC/>
</AAA>
</root>
```

I-3 - @* : l'axe attribute

L'arobase, souvent utilisée dans les prédicats, permet d'abrégé l'axe attribute.

@* ou attribute::*

```
<root>
  <AAA status="good">
    <BBB>
      <CCC status="bad" />
    </BBB>
  </AAA>
  <AAA>
    <BBB status="good" />
  </AAA>
  <AAA>
    <CCC type="empty" />
  </AAA>
</root>
```

I-4 - .. : l'axe parent

Le double point permet d'abrégé l'axe parent.

CCC/.. ou CCC/parent::*

```
<root>
  <AAA>
    <BBB>
      <CCC />
    </BBB>
    <BBB>
      <CCC />
    </BBB>
  </AAA>
  <AAA>
    <BBB />
  </AAA>
  <AAA>
    <CCC />
  </AAA>
</root>
```

I-5 - . : l'axe self

Le point permet d'abrégé l'axe self.

AAA/child::*/. ou AAA/chlid::* /self::* ou encore AAA/chlid::*

```
<root>
  <AAA>
    <BBB>
      <CCC />
    </BBB>
    <BBB>
      <CCC />
    </BBB>
  </AAA>
  <AAA>
    <BBB />
  </AAA>
  <AAA>
    <CCC />
  </AAA>
</root>
```

II - Les prédicats courts

Si vous avez besoin d'informations sur les [contextes XPath](#)

Un prédicat est une caractéristique ajoutée à un contexte XPath.

Au moment de la concrétisation du contexte, cette caractéristique est évaluée. Elle peut être vérifiée, ou pas.

On est donc dans une logique booléenne (oui/non), d'où le nom de [prédicat](#). De ce fait, un prédicat XPath devrait toujours être de type booléen.

Cependant, certaines notations abrégées permettent de spécifier des prédicats dont le type est différent

II-1 - Les prédicat numérique

Un prédicat numérique spécifie une position.

AAA/BBB[1] ou AAA/BBB[position()=1]

```
<root>
  <AAA>
    <BBB>
      <CCC/>
    </BBB>
  <BBB>
    <CCC/>
  </BBB>
</AAA>
<AAA>
  <BBB/>
</AAA>
<AAA>
  <CCC/>
</AAA>
</root>
```

II-2 - Les prédicats de type noeud

Un prédicat de type node-set spécifie l'existence du node-set.

Il n'y a d'ailleurs pas de fonction XPath qui permette de tester l'existence d'un noeud autrement

[@]

```
<root>
  <AAA status="good">
    <BBB>
      <CCC status="bad" />
    </BBB>
  </AAA>
  <AAA>
    <BBB status="good" />
  </AAA>
  <AAA>
    <CCC type="empty" />
  </AAA>
</root>
```

II-3 - Les prédicats implicites

Quand on précise directement le nom d'une balise, cela équivaut à mettre un prédicat avec name()

CCC/ancestor::AAA ou CCC/ancestor::*[name()='AAA']

```
<root>
  <AAA>
    <BBB>
      <CCC/>
    </BBB>
  </AAA>
  <AAA>
    <BBB/>
  </AAA>
  <AAA>
    <CCC/>
  </AAA>
</root>
```