



# Modélisation XDM

---

<http://www.w3.org/TR/xpath-datamodel/>



# Objectifs

---

- Proposer une représentation abstraite du document pour pouvoir
  - Localiser – atteindre des fragments de ce document
  - Traiter les fragments localisés
    - Tout type de manipulation nécessite la localisation préalable des fragments sur lesquels elle s'applique  
Extraction / Transformation / Construction

→ **Représentation arborescente**



# Document exemple

## 6 – COL D'IZOARD

*Cette montée pédestre, au haut lieu du cyclotourisme, vous évite les lacets de la route et, du fond du vallon du torrent d'Izoard, vous permet d'admirer, loin des foules et de la motorisation, les rochers déchiquetés de la Casse Déserte (cargneules).*

**Brunissard : 1760 m                      Cotation : 1**

**Col d'Izoard : 2360 m – 2 h Dénivelée : 600 m**

**Montée :** de Brunissard remonter (NW) la route de l'Izoard, D 902, jusqu'au premier virage de l'entrée des bois ; la quitter, à gauche, pour traverser (N) le groupe de chalets de La Draye. Continuer (N) par le sentier balisé GR 58 qui remonte le ravin du torrent du Col d'Izoard. Juste avant d'arriver à la base des éboulis de la Casse Déserte, le GR 58 monte à l'Est et notre itinéraire continue (N) dans le thalweg, au pied de la Casse Déserte, puis s'élève dans les pentes Sud du col, recoupant la route D 902, pour aboutir au Col d'Izoard (2360 m). Stèle et Pavillon du Cyclotourisme.

**Descente :** sur Brunissard par l'itinéraire de montée.

Sur Cervières, par le refuge d'Izoard et la route D 902, sur le versant Nord du col.

# Document XML

```
<guide>...
```

```
  <itinéraire numéro="6">
```

```
    <nom>Col d'Izoard</nom>
```

```
    <commentaire>Cette montée pédestre, au haut lieu du cyclotourisme, vous évite les lacets de la route et, du fond du vallon du torrent d'Izoard, vous permet d'admirer, loin des foules et de la motorisation, les rochers déchiquetés de la Casse Déserte (cagneules).</commentaire>
```

```
    <cotation>1</cotation>
```

```
    <départ>
```

```
      <lieu>Brunissard</lieu>
```

```
      <altitude>1760</altitude>
```

```
    </départ>
```

```
    <arrivée>
```

```
      <lieu>Col d'Izoard</lieu>
```

```
      <altitude>2360</altitude>
```

```
    </arrivée>
```

```
    <temps unité="heure">2</temps>
```

```
    <montée>De <lieu>Brunissard<lieu> remonter (NW) la route de l'Izoard, <lieu>D 902</lieu>, jusqu'au premier virage de l'entrée des bois ...</montée>
```

```
    <descente> Sur <lieu>Brunissard<lieu> par l'itinéraire de montée. Sur Cervières, par le <lieu>refuge d'Izoard<lieu> et la route <lieu>D 902</lieu>, sur le versant Nord du col.</descente>
```

```
  </itinéraire>
```

```
...
```

```
</guide>
```

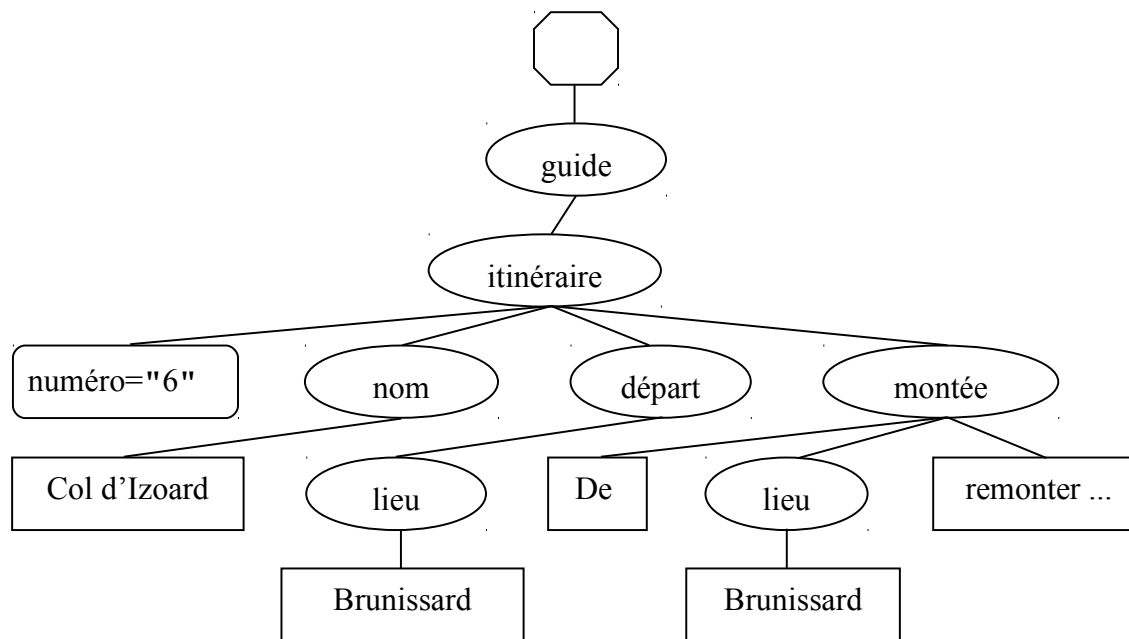


# Nœuds (1)

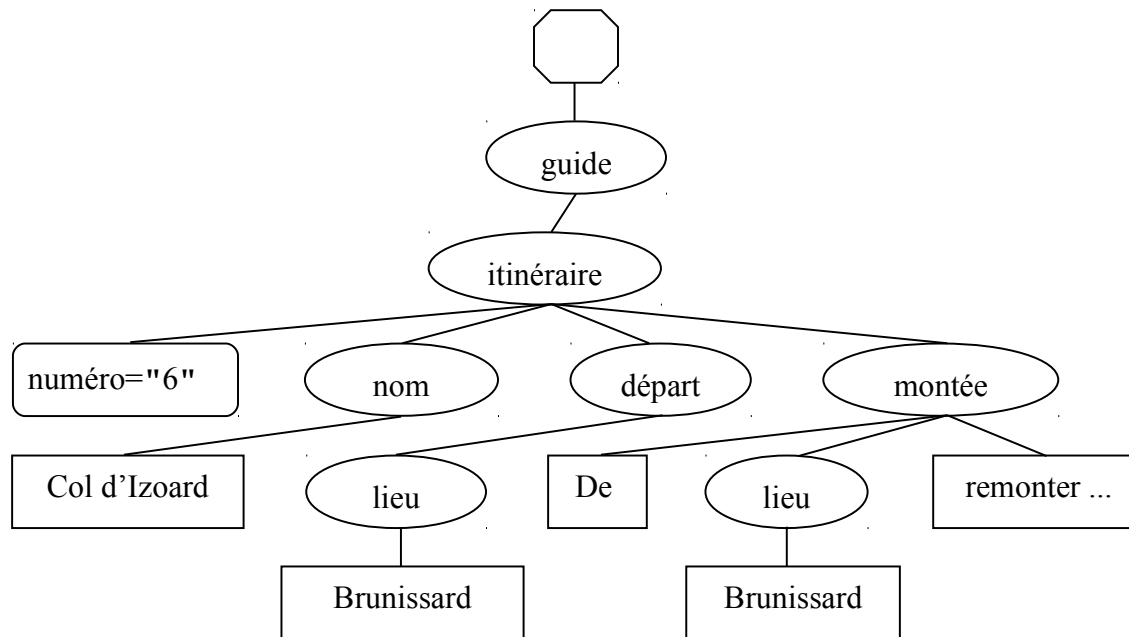
---

- L'arbre du document est formé par des nœuds
- Il existe 7 sortes de noeuds
  - *Nœud document*
  - *Nœud élément*
  - *Nœud texte*
  - *Nœud attribut*
  - Nœud espace de noms
  - Nœud instruction de traitement
  - Nœud commentaire

# Illustration : un arbre de document



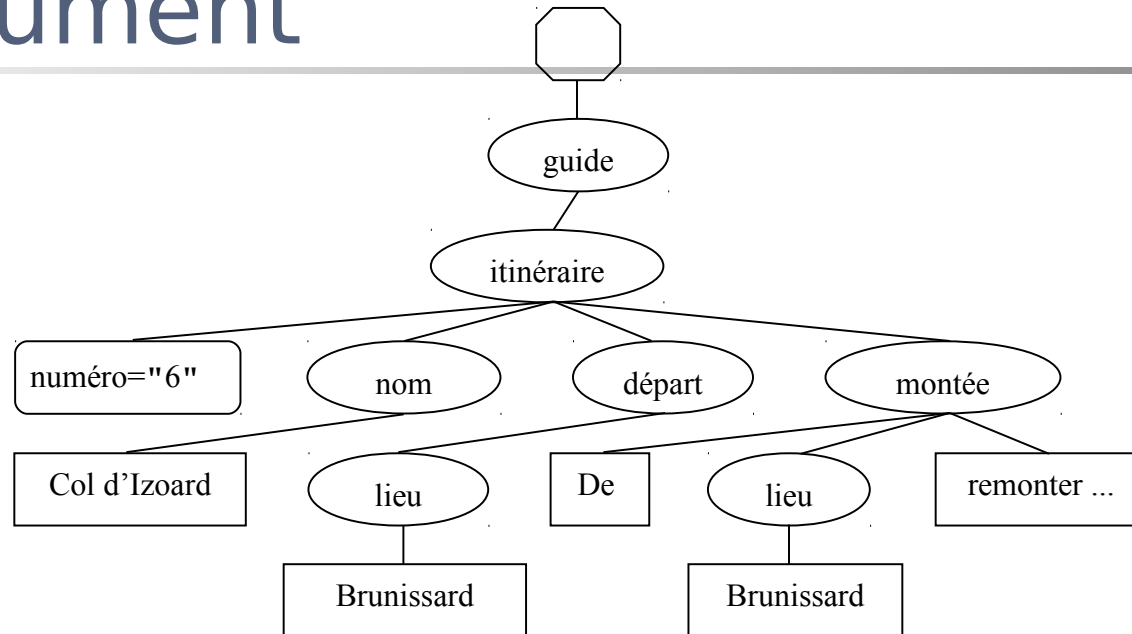
# Illustration : un arbre de document



Le **nœud document** constitue la racine de l'arbre d'un document

- il a un fils unique qui est un nœud élément
- Ce nœud élément représente l'élément de niveau supérieur du document (le premier dans l'ordre de lecture)

# Illustration : un arbre de document

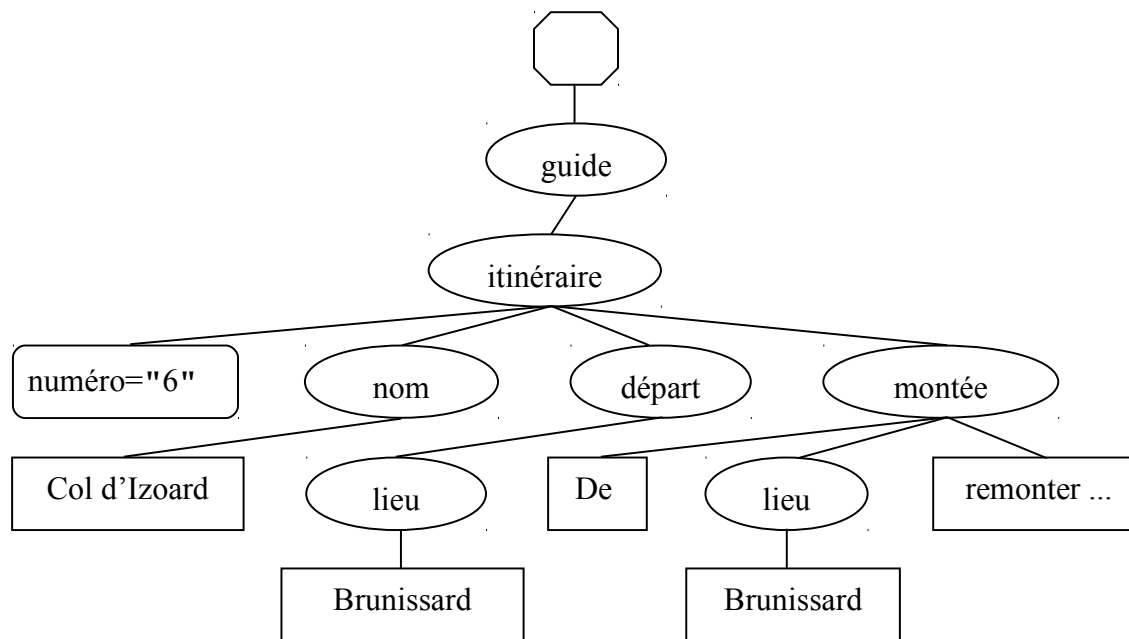


## Un nœud élément

- est étiqueté par le nom de l'élément qu'il représente
- a pour fils les nœuds représentant les attributs, les éléments et les fragments de texte qui le constituent
- parmi ces fils, les nœuds éléments et les nœuds textes sont appelés ses enfants
  - Impossible d'avoir deux nœuds enfant consécutifs qui sont des nœuds texte
- ces enfants sont ordonnés selon l'ordre de lecture du document



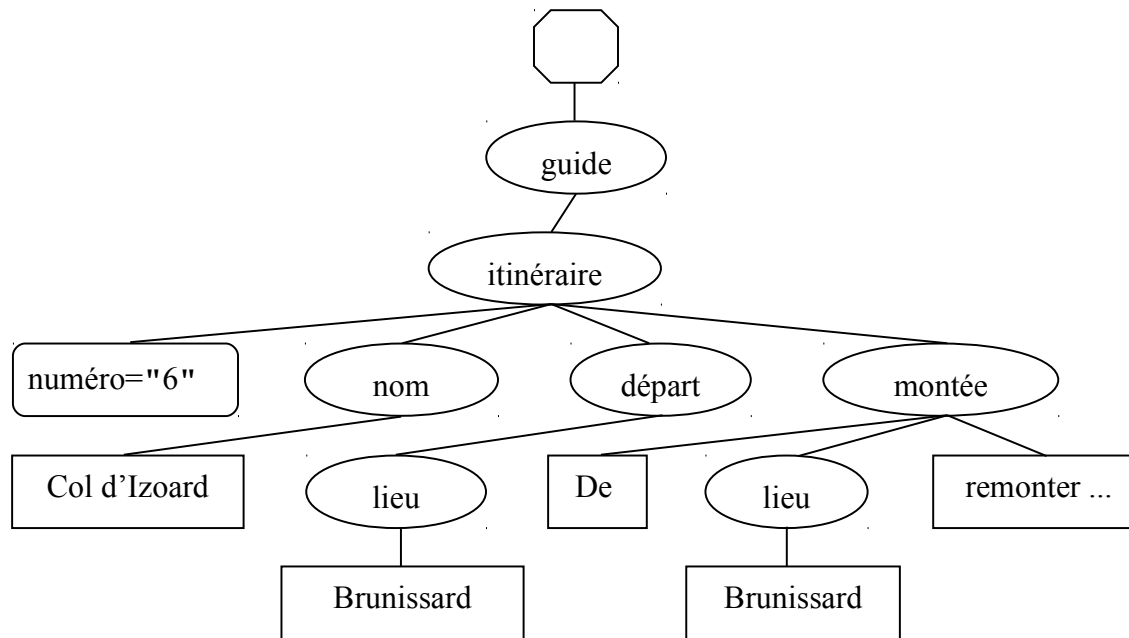
# Illustration : un arbre de document



## Un **nœud attribut**

- est étiqueté par le nom et la valeur de l'attribut qu'il représente
- a pour père un nœud élément et n'a pas de nœud fils

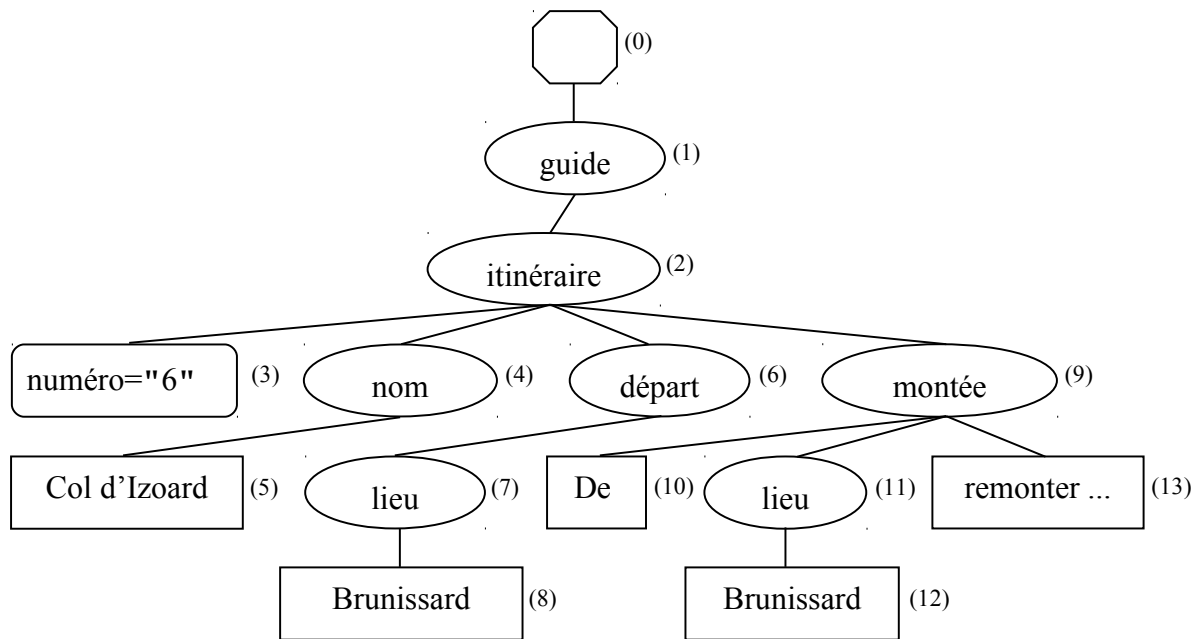
# Illustration : un arbre de document



## Un **nœud texte**

- est étiqueté par le fragment de texte qu'il représente
- a pour père un nœud élément et n'a pas de nœud fils

# Illustration



L'ordre du document : ordre de lecture de ses constituants représentés par des nœuds



# Nœuds

---

- Un nœud a un identifiant unique
- Un nœud a une valeur textuelle
  - nœud document : valeur textuelle de son nœud fils (l'élément du document)
  - nœud élément : concaténation, dans l'ordre du document, des valeurs textuelles de ses nœuds enfants (éléments et textes mais pas attributs)
  - nœud texte : fragment de texte qu'il représente
  - nœud attribut : valeur de cet attribut

la valeur du nœud (9) est "De Brunissard remonter... "  
la valeur du nœud (3) est "6"  
la valeur du nœud (5) est "Col d'Izoard"



# Séquences

---

- Une instance du modèle est une **séquence**
  - collection ordonnée de 0 ou plusieurs **items**
- Un item est soit un **nœud**, soit une **valeur atomique**
- Une valeur atomique est une instance d'un type atomique (cf.XSchema)
- Remarques
  - Une séquence de 0 item est appelée **séquence vide**
  - Une séquence de 1 item est appelée **séquence singleton**
  - Il y a équivalence entre un item et une séquence singleton
  - Les séquences sont plates
    - une séquence ne peut pas être un élément d'une séquence



# XPath: Localisation de composants d'un document XML

---

<http://www.w3.org/TR/xpath/>

<http://www.w3.org/TR/xpath20/>

<http://www.w3.org/TR/xpath-functions/>

Jacques Le Maitre, *Description et manipulation de documents XML*, supports de cours <http://lemaitre.univ-tln.fr/cours.htm>



# Localiser

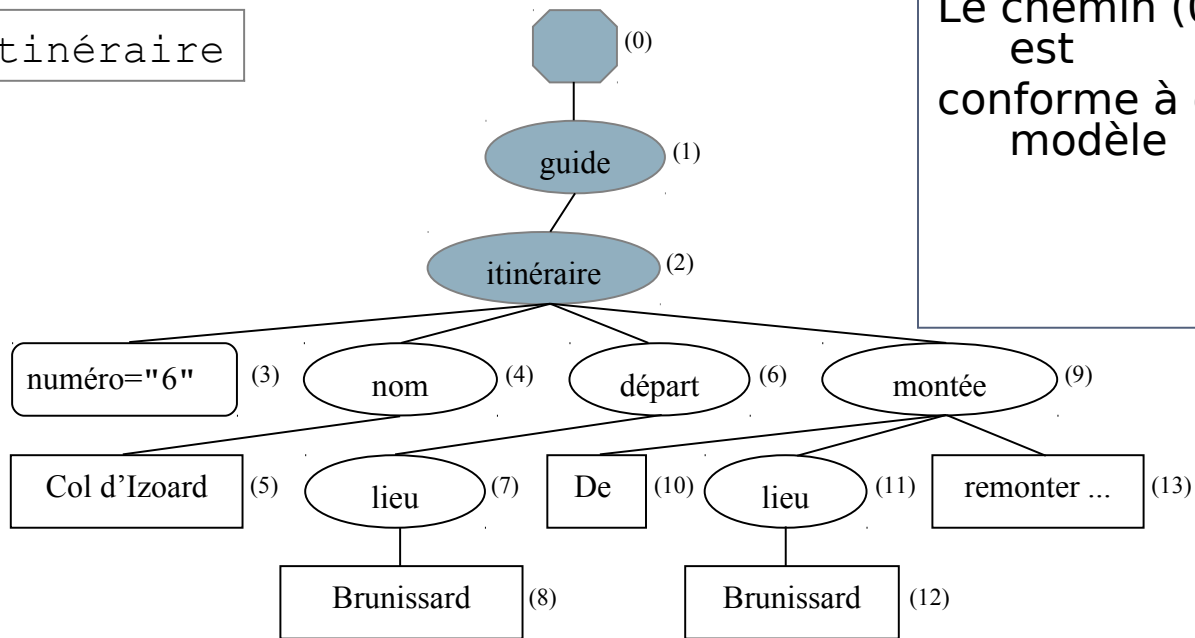
---

- Localiser des fragments de documents - des nœuds - dans un arbre XML
- En atteignant et en contraignant :
  - un composant même
    - un élément `Itinéraire` ou un attribut `numéro`
  - son modèle de contenu
    - un itinéraire contenant le mot `Brunissard` ou un itinéraire contenant une cotation donnée
  - le chemin à suivre pour atteindre ce composant en tenant (éventuellement) compte de son environnement
    - la descente et la montée se font par le même itinéraire

# Exemples d'expressions XPath

(écriture abrégée)

`/guide/itinéraire`



Le chemin (0, 1, 2)  
est  
conforme à ce  
modèle

- *Décrire* un modèle de chemin dans l'arbre d'un document : **expression de chemin**
- A partir d'un nœud de départ dit nœud contexte, *sélectionner* les nœuds atteints en suivant tous les chemins conformes à ce modèle,
- Exploiter les relations généalogiques existant entre les nœuds qui composent le document

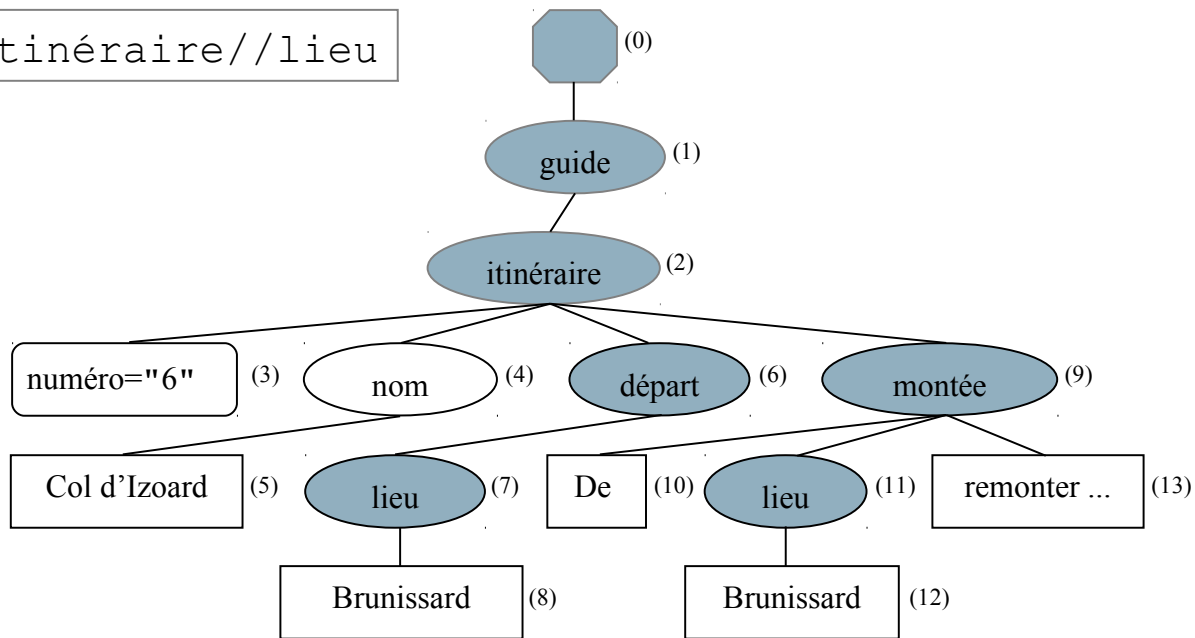


# Exemples d'expressions XPath

## XPath

(écriture abrégée)

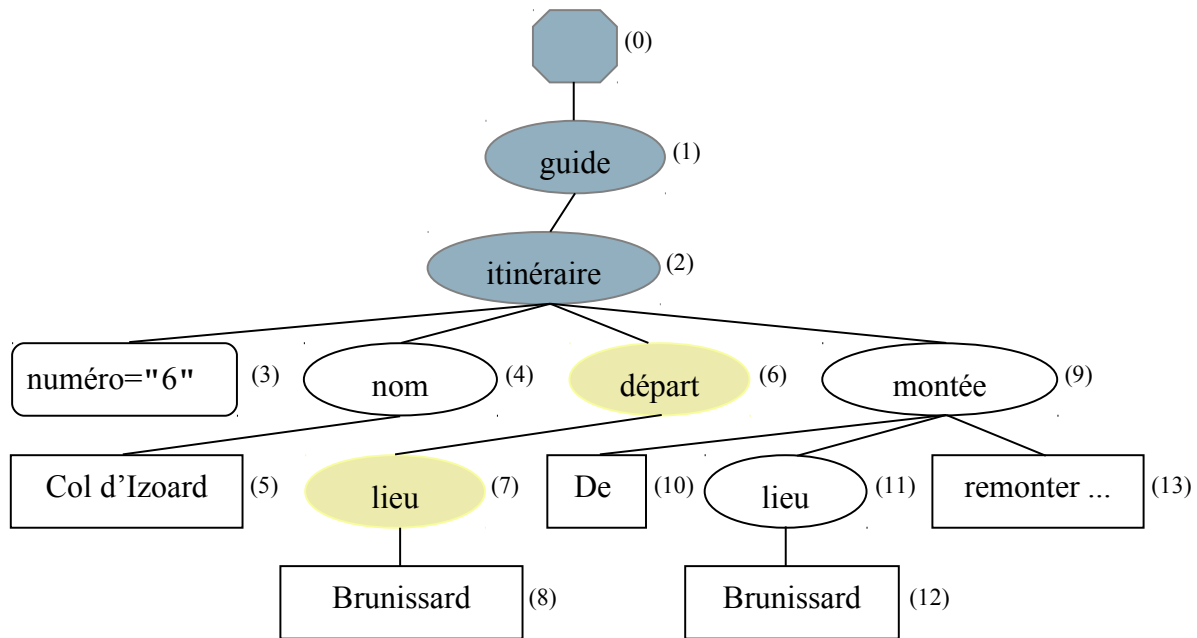
`/guide/itinéraire//lieu`



Les chemins (0, 1, 2, 6, 7) et (0, 1, 2, 9, 11) sont conformes à ce modèle

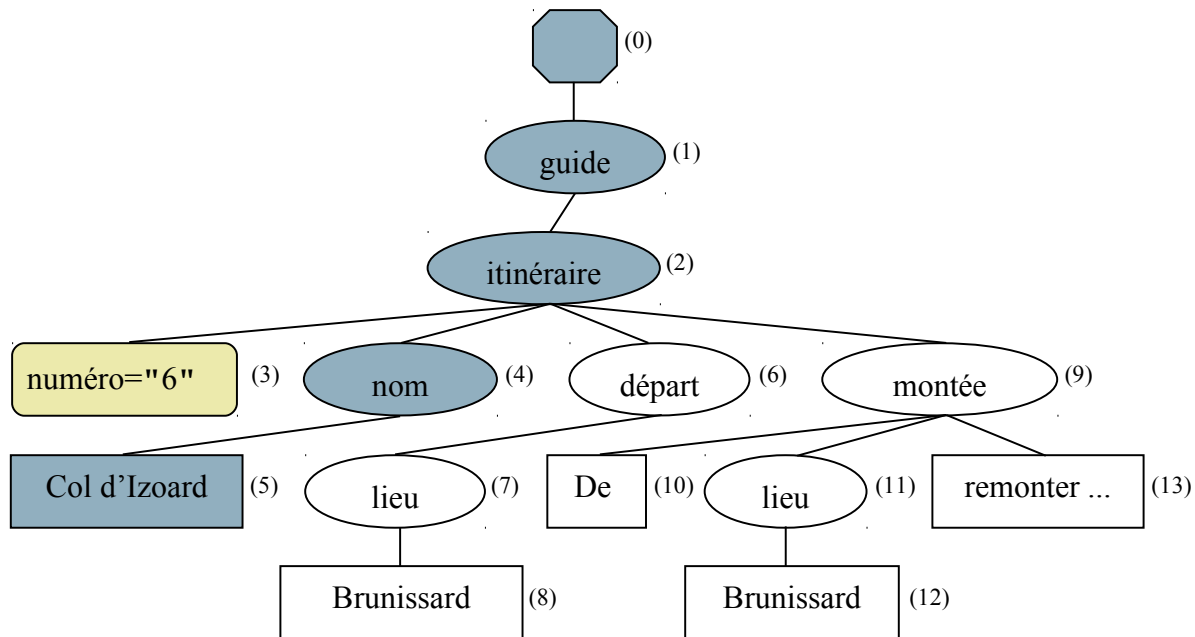
```
/guide/itinéraire[départ/lieu = "Brunissard"]
```

# Modèle de données XPath



```
/guide/itinéraire[@numéro = "6"]/nom/text()
```

# Modèle de données XPath





# XPath

---

- XPath est notamment utilisé dans :
  - *XQuery*, pour localiser un nœud précis ou un ensemble de nœuds afin de le (les) post-traiter
  - *XSLT* pour localiser un nœud précis ou un ensemble de nœuds afin de leur associer par exemple une présentation
  - *XLink* pour pointer une cible précise dans un document (XPointer)



# Expression

---

Elle est construite à partir :

- Constantes littérales (chaînes, nombres)
- Noms de variable
- Chemins de localisation
- Opérateurs (sur les nœuds, arithmétiques, etc.)
- Fonctions prédéfinies

- Elle est évaluée dans
  - un contexte statique (analyse de l'expression)
  - un contexte dynamique (évaluation de l'expression)
- Sa valeur est une instance du modèle XDM : une séquence de 0 ou plusieurs items
  - Un item est un nœud ou une valeur atomique
  - Un item n'est pas une séquence

# Expression de chemin

$exp_1/exp_2$

- $exp_1$  doit avoir pour valeur une séquence de nœuds  $S_1$
- $exp_2$  est évaluée pour chaque nœud  $node$  de  $S_1$ 
  - Nœud contexte:  $node$
  - Position contexte: position de  $node$  dans la séquence  $S_1$
  - Taille contexte : longueur de  $S_1$
- Chemin relatif
  - point de départ : le nœud contexte est un nœud quelconque du document
- Chemin absolu
  - point de départ : le nœud contexte est la racine du document

`exp1/exp2`

`/guide/itinéraire[position()=2]`

# Description d'un pas de l'expression

`axe::test_de_nœud [p1]...[pn]`

- Le pas est décrit par:
  - un axe
  - un test de nœud
  - une suite de prédicats (peut-être vide)
- Sa valeur est la séquence de nœuds atteints à partir du nœud contexte en suivant l'axe, vérifiant le test de nœud et les prédicats successifs



# Axe

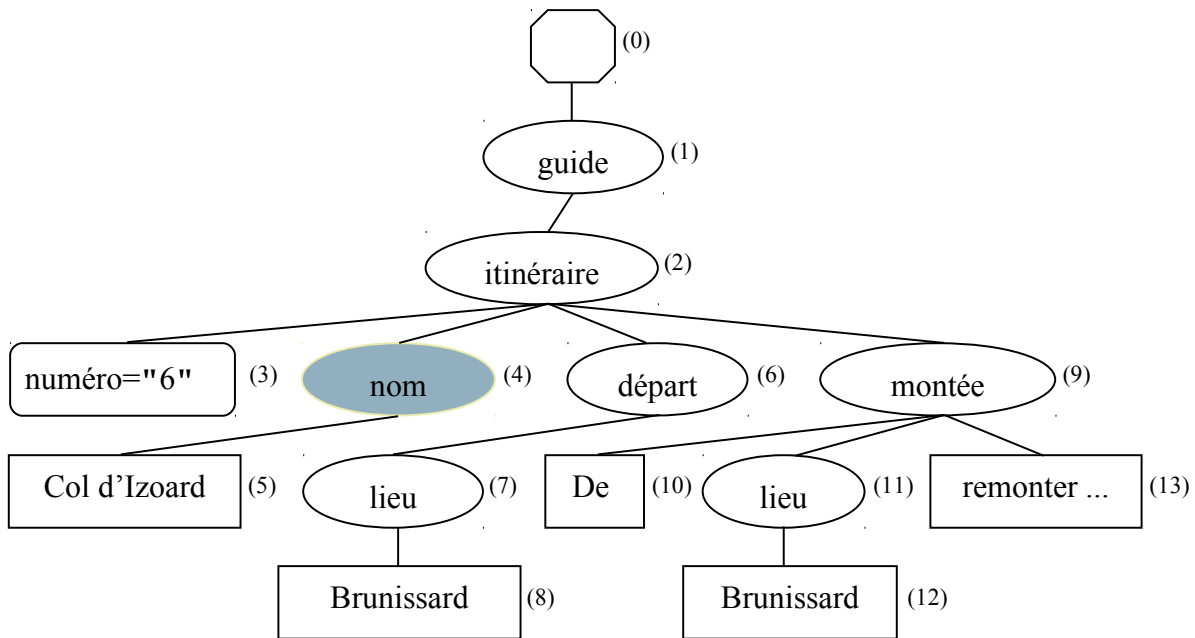
---

- Précise la direction à suivre pour atteindre les nœuds suivants
  - Hiérarchie directe, indirecte entre les nœuds
  - Position relative des nœuds les uns par rapport aux autres
- Un axe a une sorte de nœud principal
  - `attribut`
  - `element`
- Un axe a un sens (selon l'ordre du document)
  - **Sens avant**
    - `child, descendant, descendant-or-self`
    - `following, following-sibling`
  - **Sens arrière**
    - `parent, ancestor, ancestor-or-self,`
    - `preceding, preceding-sibling`



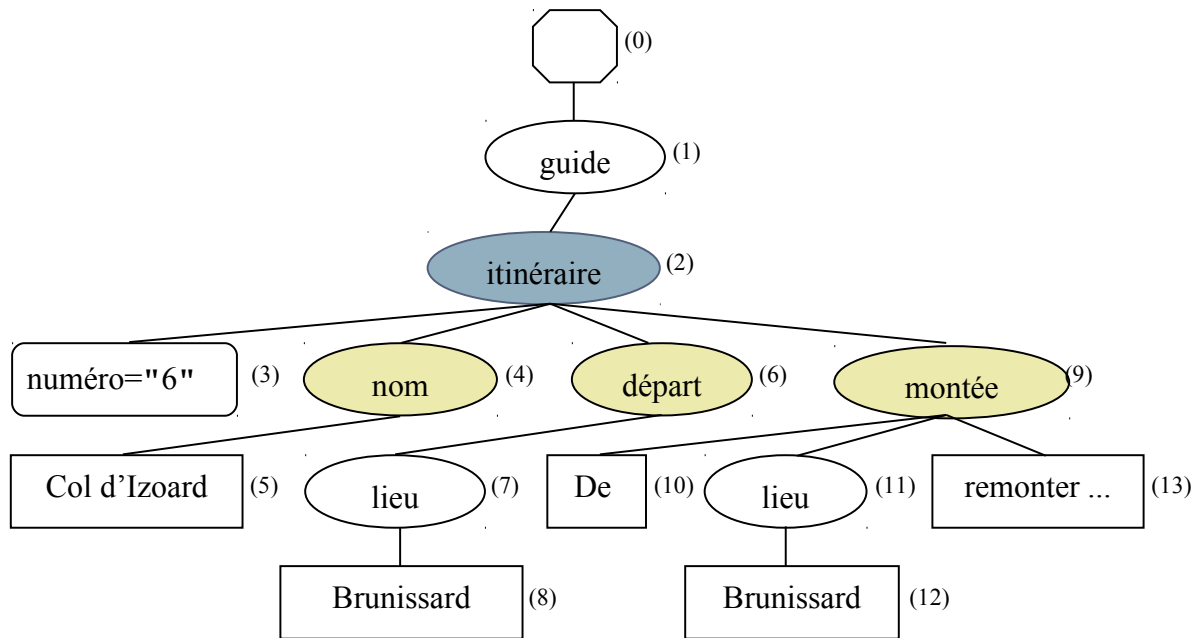
Nœud contexte : (4)  
Nœuds sélectionnés : (4)

Self::\*



Nœud contexte : (2)  
Nœuds sélectionnés : (4),(6),(9)

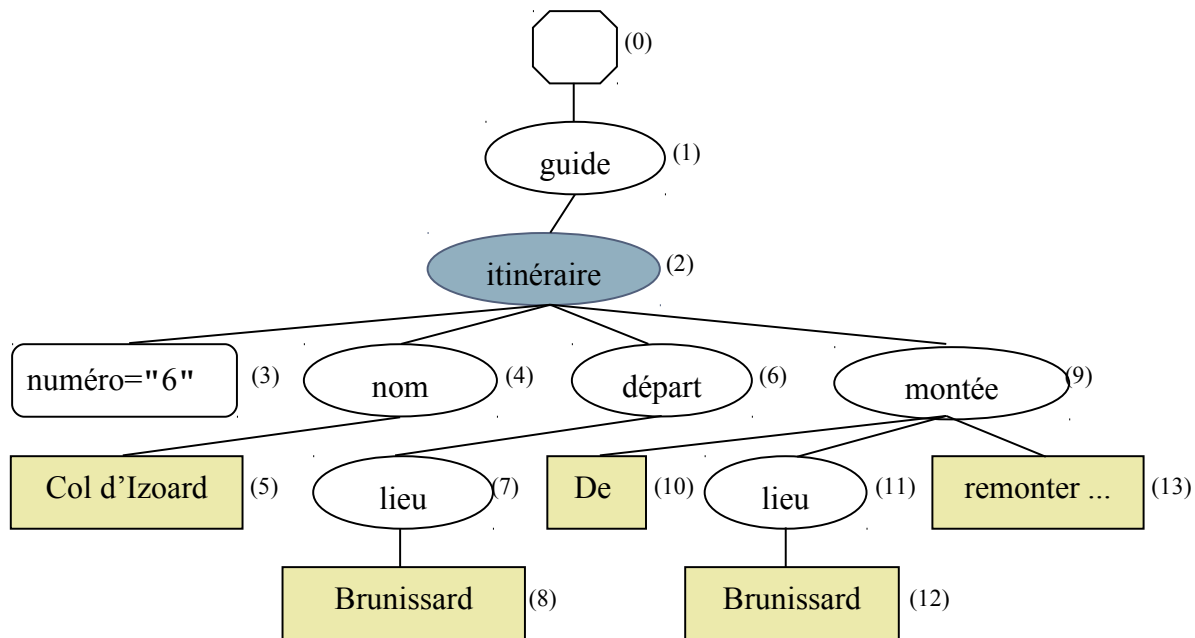
Child::\*



Nœud contexte : (2)

Nœuds sélectionnés : (5),(8),(10),(12),(13)

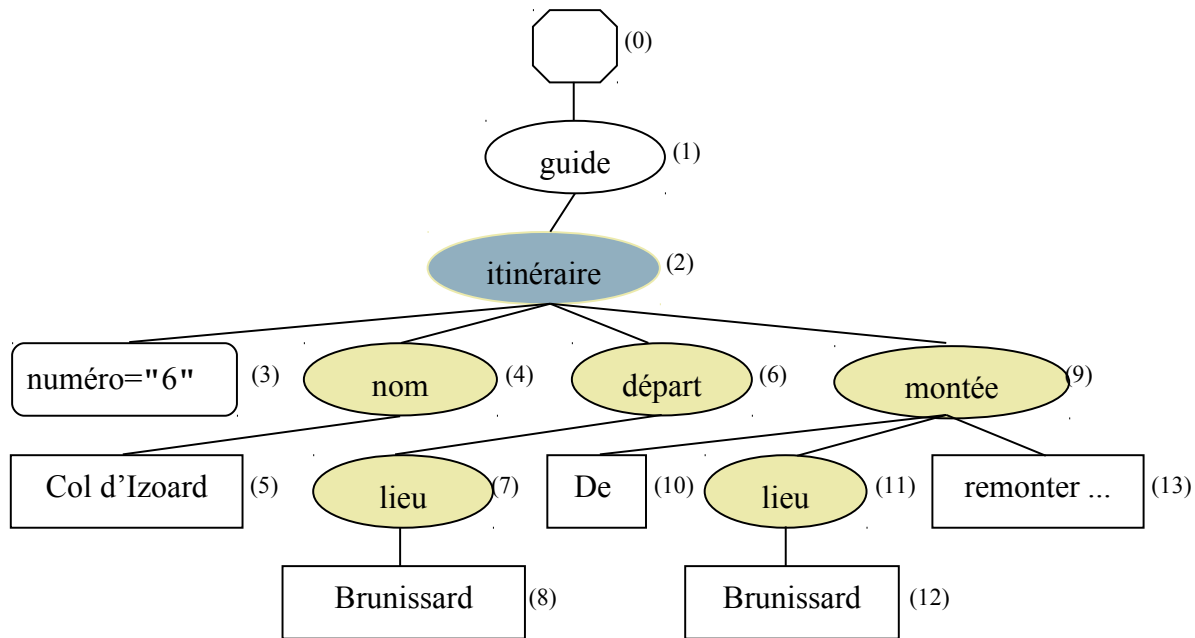
# Descendant::text()



Nœud contexte : (2)

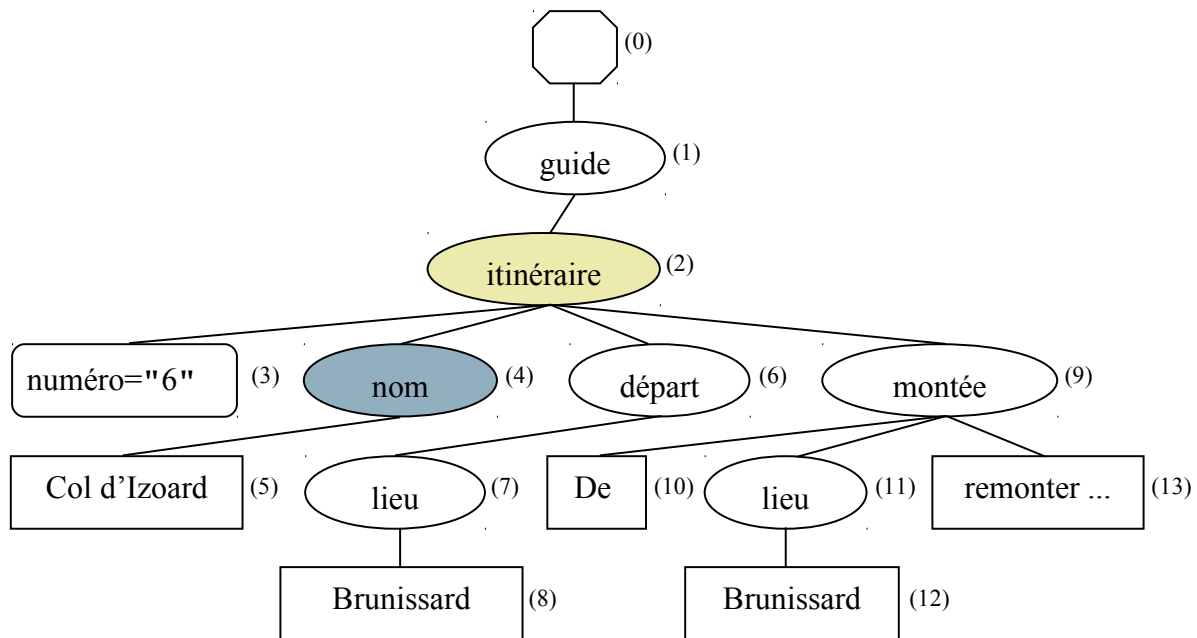
Nœuds sélectionnés : (2),(4),(6),(7),(9),(11)

# Descendant-or-self::\*



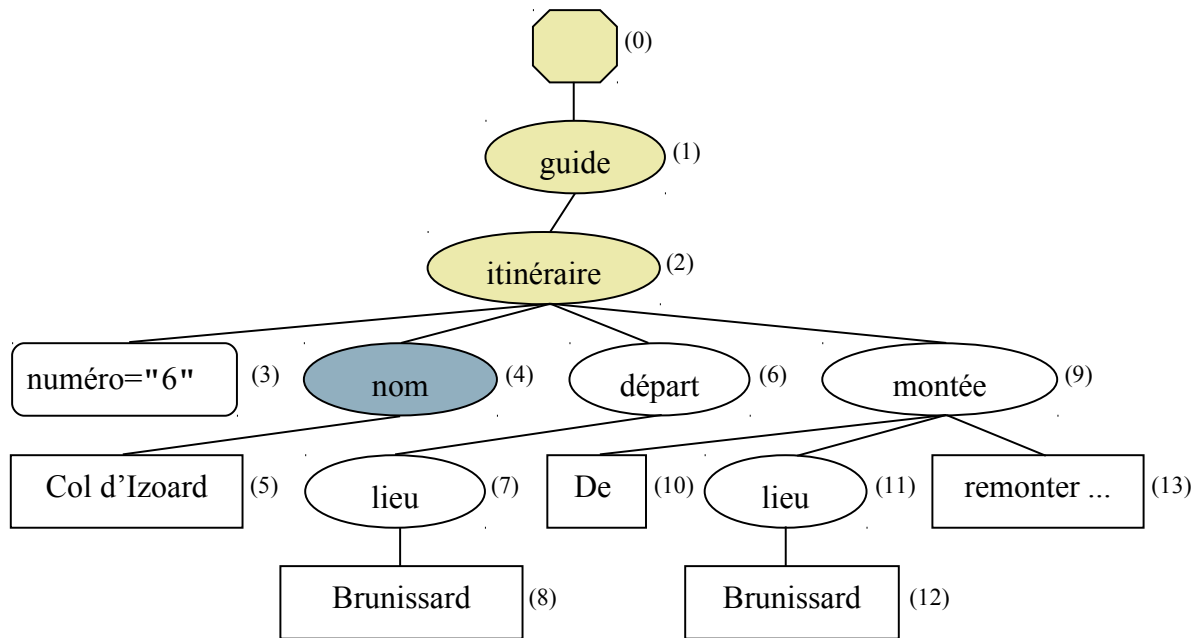
Nœud contexte : (4)  
Nœuds sélectionnés : (2)

Parent::\*



Nœud contexte : (4)  
Nœuds sélectionnés : (2),(1),(0)

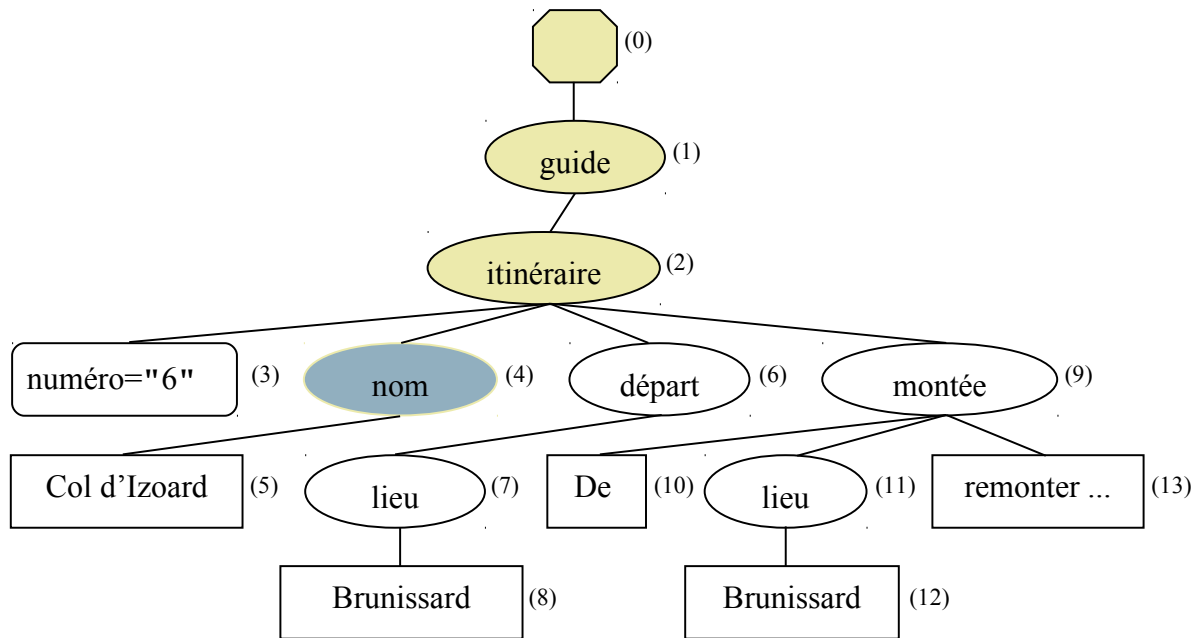
# Ancestor::\*



Nœud contexte : (4)

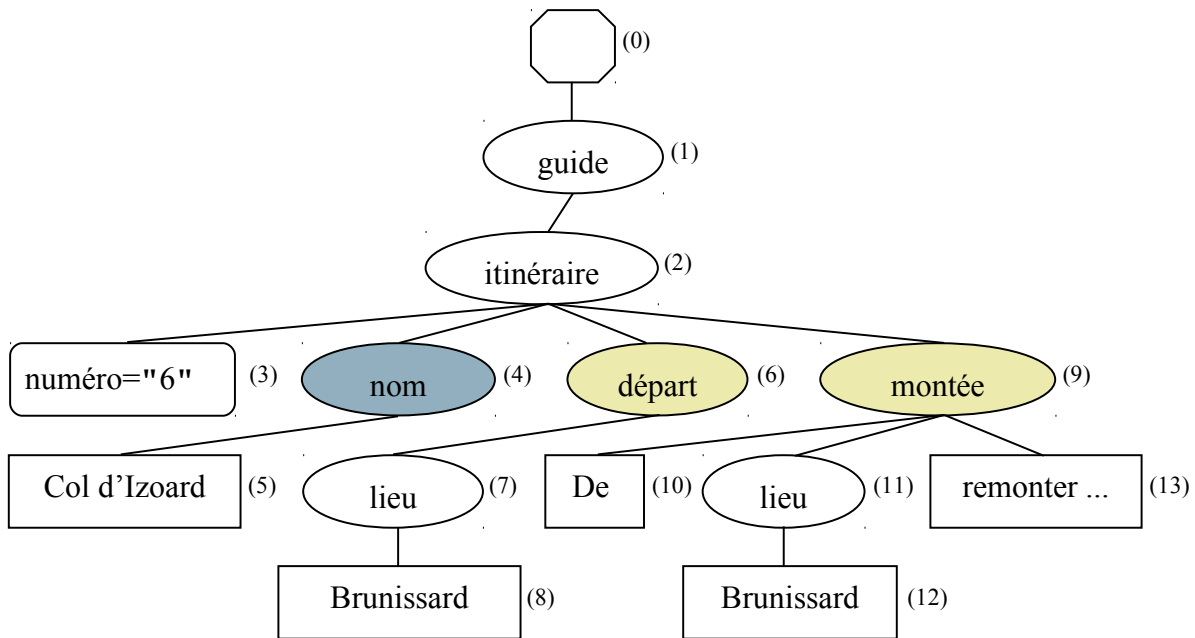
Nœuds sélectionnés : (4),(2),(1),(0)

# Ancestor-or-self::\*



Nœud contexte : (4)  
Nœuds sélectionnés : (6), (9)

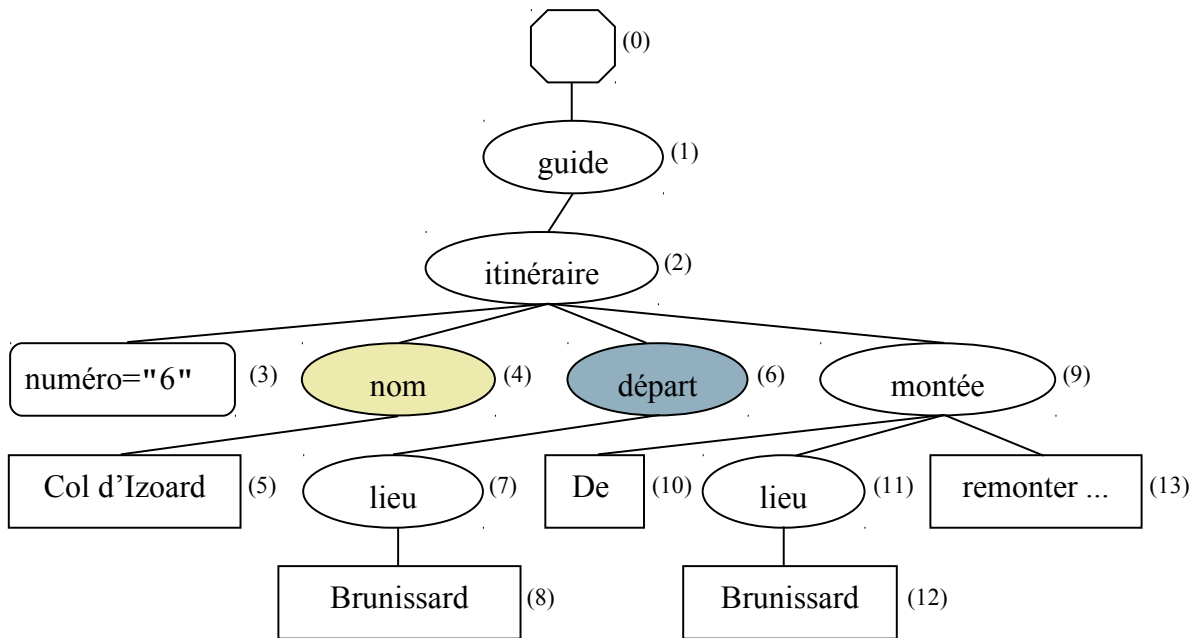
# Following-sibling::\*





Nœud contexte : (6)  
Nœuds sélectionnés : (4)

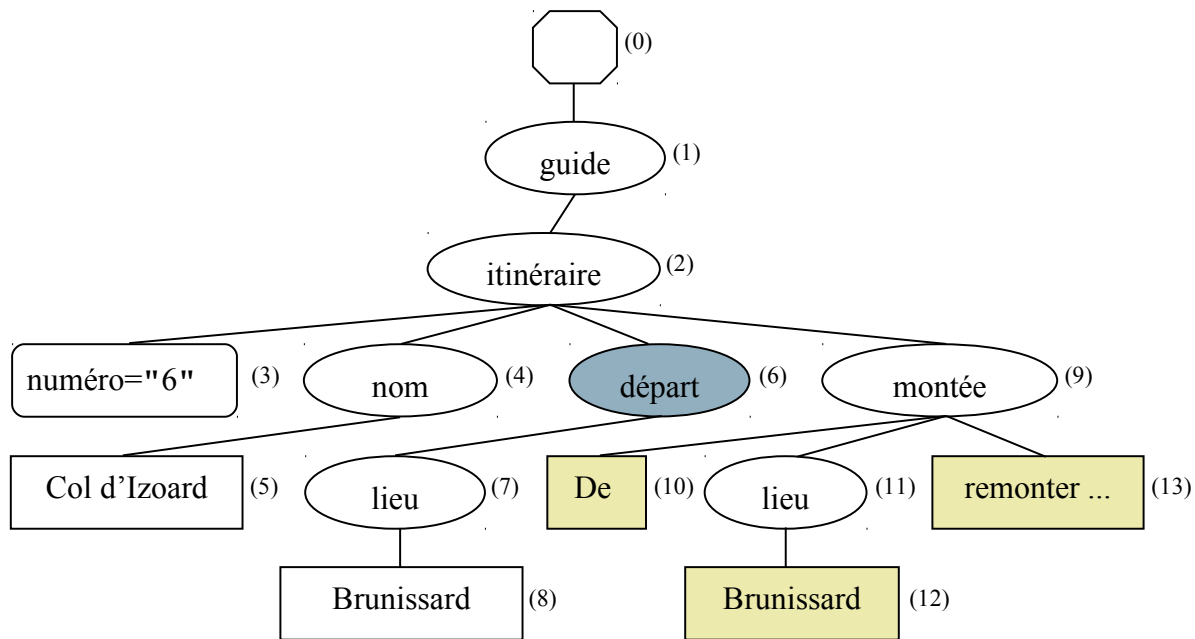
# Preceding-sibling::\*



Nœud contexte : (6)

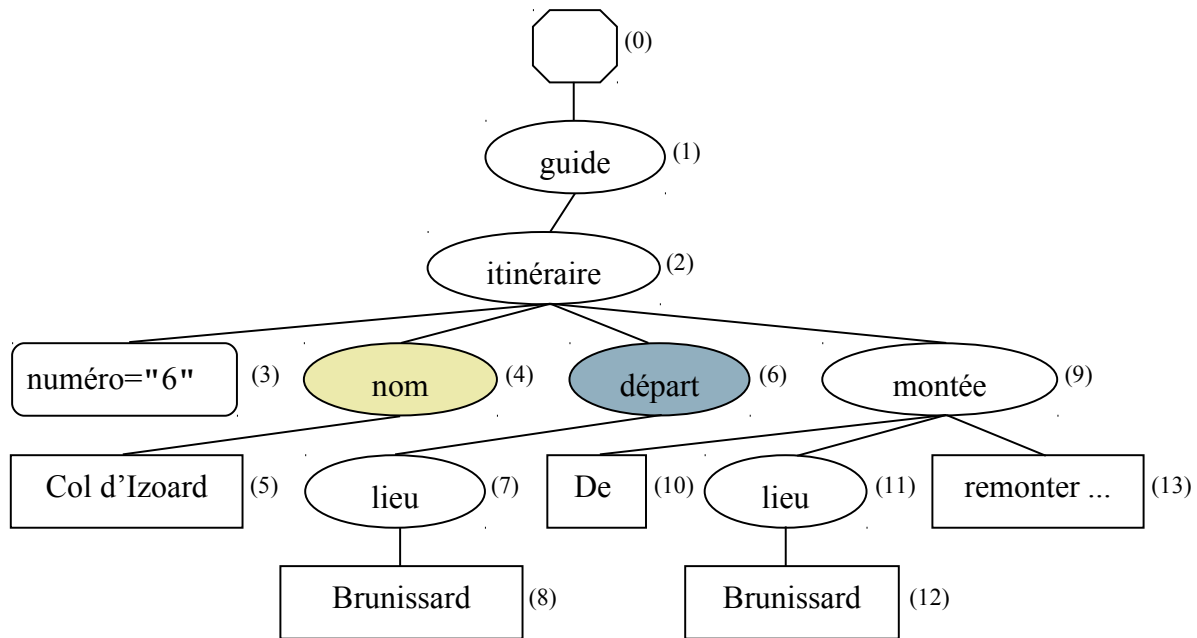
Nœuds sélectionnés : (10),(12),(13)

# Following::text()



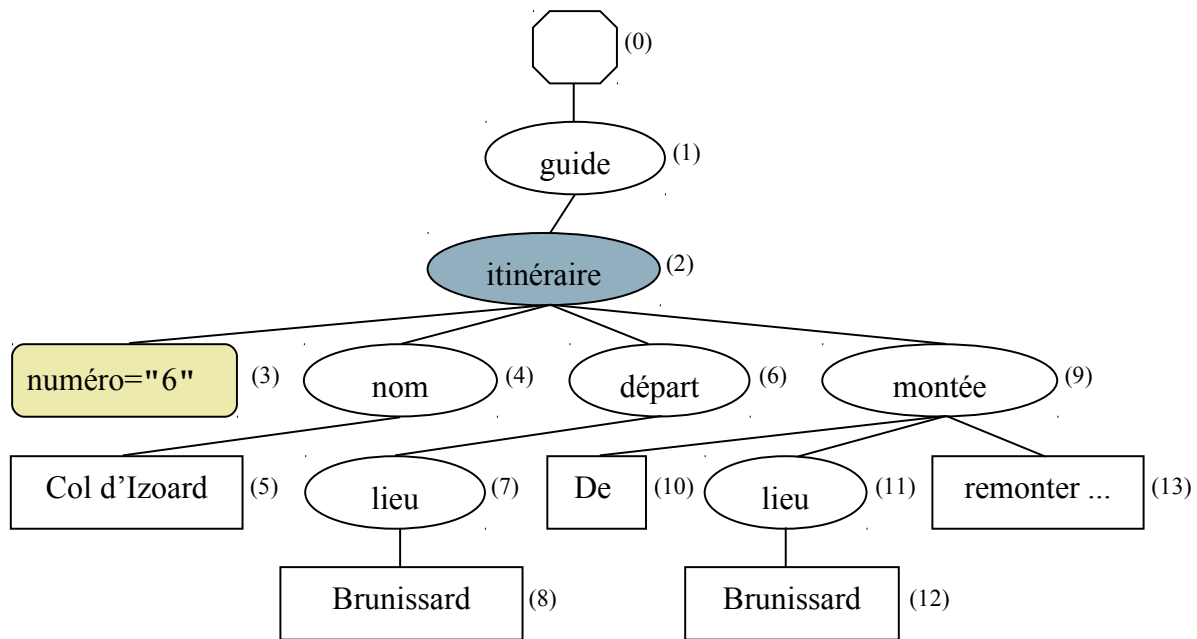
Nœud contexte : (6)  
Nœuds sélectionnés : (4)

# Preceding::\*



Nœud contexte : (2)  
Nœuds sélectionnés : (3)

# Attribute::\*





# Test de nœud

---

```
child::titre  
child:*  
child::text()  
child::node()
```

- Sélectionne la séquence des nœuds de l'axe d'une certaine sorte

<code>n</code>	tous les nœuds de nom <i>n</i>
<code>*</code>	tous les nœuds conformes avec la sorte de l'axe
<code>node()</code>	tous les nœuds
<code>text()</code>	tous les nœuds texte

# Prédicat :

## filtrer des séquences de noeuds

---

- Condition à tester sur la séquence de nœuds atteints en suivant l'axe et vérifiant le test de nœud
- Expression booléenne construite à partir d'expressions de chemin et/ou de fonctions prédéfinies
  - `fn:Position()`
    - retourne le rang du nœud à tester (sa valeur est la position contexte )
  - `fn>Last()`
    - retourne le rang du dernier nœud à tester (sa valeur est la taille contexte)
  - `fn:Count(liste_de_nœuds)`
    - dénombre le nombre de *nœuds* de *liste\_de\_nœuds*



# Prédicats : exemples

---

- L'axe de filtrage par défaut est `child`

```
child::titre [position()=2]
child::section[count(para)>2]
```

- On considère
  - la séquence de nœuds (4), (6), (9)
  - un axe avant

`[position() = 1]` est vérifié par (4) `[1]`

`[position() = last()]` est vérifié par (9) `[last()]`

`[lieu = "Brunissard"]` est vérifié par (9) et (6)

`[lieu]` est vérifié par (6) et (9)



# Une syntaxe abrégée

---

`child::test_de_nøud`  
**est équivalent à**  
`test_de_nøud`

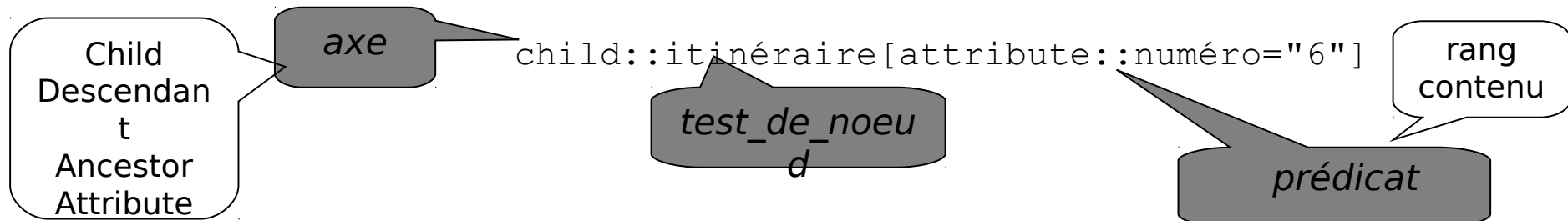
`attribute::test_de_nøud`  
**est équivalent à**  
`@test_de_noeud`

`descendant-or-self::test_de_nøud`  
**est équivalent à**  
`//test_de_nøud`



```
/descendant-or-self::itineraire/child::nom[attribute::numero="6"]
```

# Récapitulatif



```
//itineraire/nom/text()
```

- Sélectionner l'ensemble des nœuds
  - atteints en suivant tous les chemins conformes à un modèle
  - à partir de chaque nœud (dit nœud contexte) de l'ensemble des nœuds atteints au pas précédent



# Evaluation d'un pas de l'expression

---

`axe::test_de_nœud [exp1] ... [expn]`

- A partir de  $S$ , une séquence de nœuds
- Pour chaque nœud de  $S$  (**nœud contexte**)
  - On calcule la séquence de nœuds  $N$  sélectionnés par l'axe puis le *test de nœud*
  - On calcule la sous séquence  $N_1$  de  $N$  vérifiant  $exp_1$
  - On calcule la sous séquence  $N_2$  de  $N_1$  vérifiant  $exp_2$
  - ...
  - On calcule la sous séquence  $N_n$  de  $N_{n-1}$  vérifiant  $exp_n$
- Résultat
  - l'union des séquences  $N_n$  des nœuds atteints à partir de chaque nœud de  $S$*



# Evaluation d'un chemin

---

$pas_1 / \dots / pas_n$

à partir d'un nœud  $n$

Résultat = séquence des nœuds atteints

par le pas  $pas_n$

à partir de la séquence des nœuds atteints par le pas

$pas_{n-1}$

...

à partir de la séquence des nœuds atteints par le pas

$pas_1$

à partir de la séquence de nœuds composée de l'unique nœud  $n$



# Exemples d'expression XPath

---



# Exemple (deux syntaxes)

---

```
/child::guide/child::*/child::lieu/child::text()
```

```
/guide/*/lieu/text()
```



# Exemple (deux syntaxes)

---

```
/descendant-or-self::node() /  
  child::itinéraire[cotation="1"]/attribute::numéro
```

```
//itinéraire[cotation = "1"]/@numéro
```



# Exemples

---

```
//itinéraire[temps < 3][position() <= 10]
```

```
//itinéraire[contains(montée, "raide") and  
contains(montée, "éboulis) ]/commentaire
```

```
//itinéraire[.//lieu[text() = "GR 51"]]
```

```
id("6")/nom/text()
```



# Bilan

---

- XPath est un mécanisme standard de filtrage puissant et complet
- Mécanisme essentiel pour un langage d'interrogation, de manipulation
  - Un moyen de filtrer les composants du document quel que soit leur type
  - Un jeu d'opérateurs est ensuite nécessaire pour les exploiter
- Extension par opérateurs de recherche plein texte