

IV. Localisation des composants avec XPath

...

Langage XPath

- Est un **langage d'expression** (non XML)
- Standard du W3C: XPath1.0 (1999), XPath2.0 (2010), XPath3.0 (2014), XPath3.1 (2017)
- **Objectif:** exprimer des requêtes pour localiser des parties d'un document XML.
- Il est intégré dans
 - *XSLT* pour localiser un nœud précis ou un ensemble de nœuds afin de leur associer par exemple une présentation
 - *Xquery*: pour localiser un nœud précis ou un ensemble de nœuds afin de le (les) post-traiter
 - *Xlink*: pour pointer une cible précise dans un document (Xpointer)
 - *XPointer* : pointer des éléments de documents avec des expressions XPath dans les URL

Langage XPath

Exemple

```
<adresses>
  <!--Carnet d'adresses-->
  <personne id= " personnel " >
    <type classe= " famille " />
    <nom>Durand</nom>
    <prenom>Martine</prenom>
    <adr:adresse>6, rue des Magnolias
  </adr:adresse>
    anniversaire: le 6août
  </personne>
</adresses>
```

Exemple d'expression Xpath:

/child::adresses /child::personne /child::nom /child::text()

Résultat de d'expression Xpath:

Durand

Outils pour explorer XPath

- oXygen XML editor (<https://www.oxygenxml.com/>)
- XPath Visualiser
- Altova XMLSpy
- CookTop (<http://www.toocharger.com/fiches/windows/cooktop/46333.htm>)
- XML Copy Editor
- Zorba (<http://www.zorba-xquery.com/>). Il s'agit d'un interprète du langage XQuery dont XPath est un sous ensemble.
- <http://eclipsexslt.sourceforge.net> propose XPath Navigator qui permet de visualiser le résultat d'une expression XPath dans eclipse.

Plan de cette partie

1. Arbre XML: Modèle de données XDM (XML Data Model)
2. Le langage XPath
 - a) Expressions de chemin
 - b) Les axes
 - c) Les filtres
 - d) Les prédicats

Modèle de données XDM

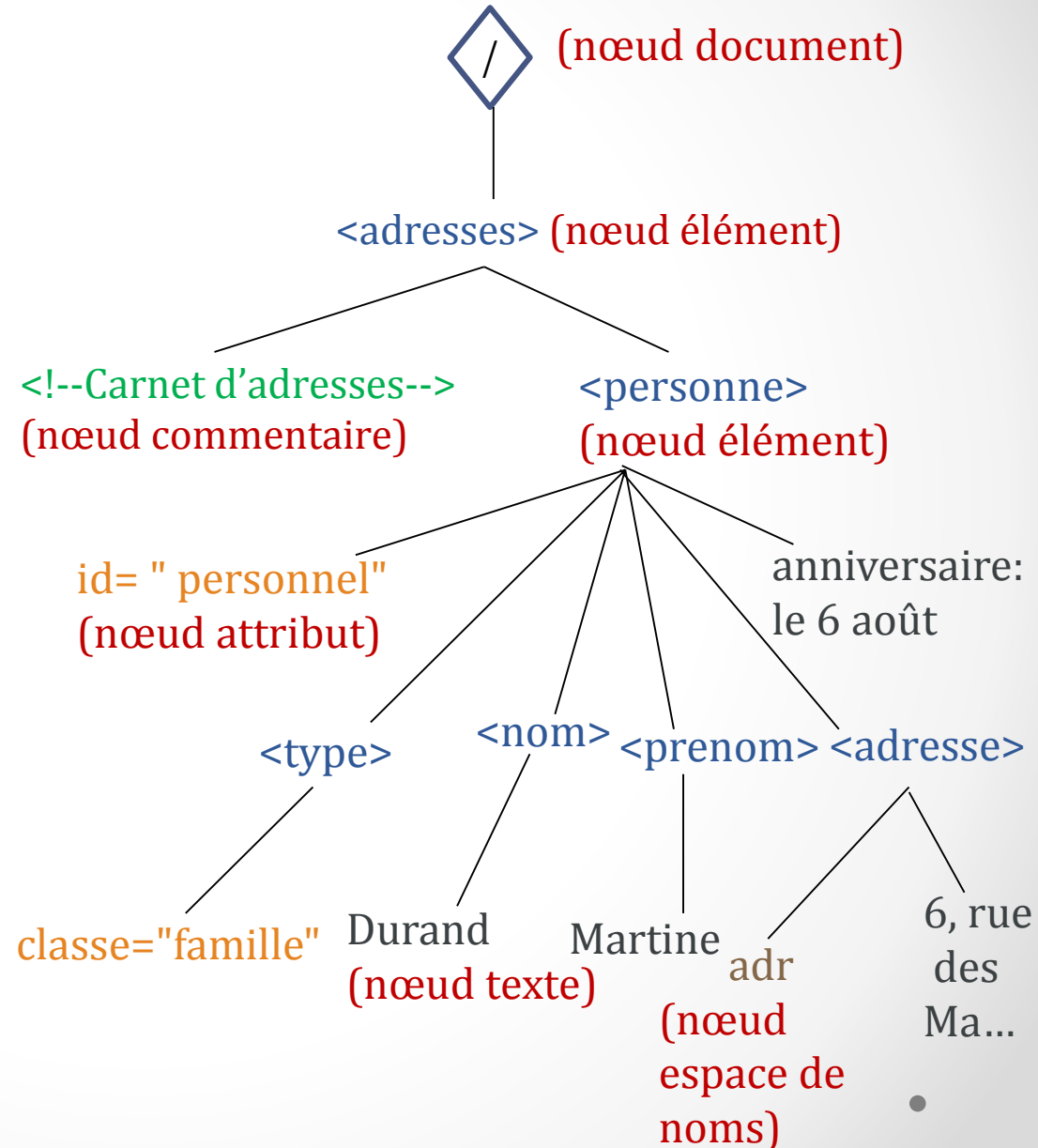
- Un modèle de représentation arborescente d'un document XML: le modèle **XDM (XML Data Model)**
- **Le modèle XDM** est commun à plusieurs langages de manipulation de données XML :
 - XPath 2.0
 - XQuery 1.0
 - XSLT 2.0

Modèle de données XDM

- Document XML: structure hiérarchique en arbre
- Avec quelques modifications:
 - **Ajout d'un nœud « document »** au dessus de l'élément de plus haut niveau
 - Les attributs et les textes sont des nœuds
 - Les commentaires et instructions de traitement sont des nœuds

Exemple

```
<adresses>
|
|  <!--Carnet d'adresses-->
|  <personne id= " personnel " >
|  |
|  |  <type classe= " famille " />
|  |  <nom>Durand</nom>
|  |  <prenom>Martine</prenom>
|  |  <adr:adresse>6, rue des Magnolias
|  |  </adr:adresse>
|  |  anniversaire: le 6août
|  |
|  </personne>
|
</adresses>
```



Nœuds

- Un document XML est une arborescence composée de Nœuds
- Il existe 7 sortes de nœuds
 - Nœud document
 - Nœud élément
 - Nœud texte
 - Nœud attribut
 - Nœud espace de noms
 - Nœud commentaire
 - Nœud instruction de traitement

Nœuds

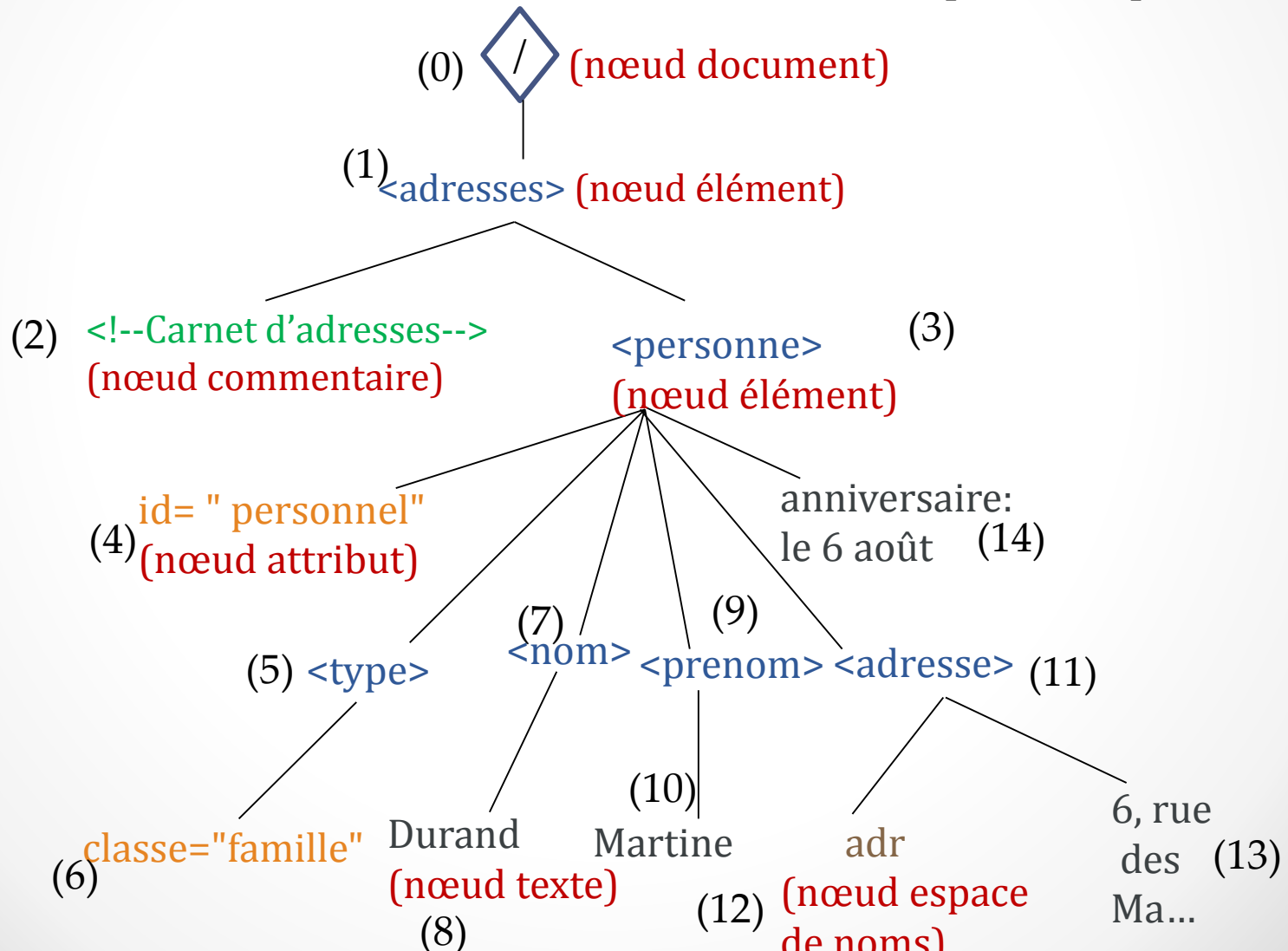
- Le **nœud document** constitue la racine de l'arbre d'un document
 - à ne pas confondre avec l'élément racine
 - Est l'ancêtre de tous les nœuds de l'arborescence XPath associée à un document XML.
 - Représente un document dans son ensemble
 - Ne possède pas d'élément parent
 - Les enfants d'un nœud document sont ses nœuds fils de type commentaire ou instruction de traitement et le nœud associé à l'élément racine
- Un **nœud élément**
 - est étiqueté par le nom de l'élément qu'il représente
 - Ses fils, les nœuds éléments, texte, commentaire ou instruction de traitement sont appelés ses enfants
 - Impossible d'avoir deux nœuds enfants consécutifs qui sont des nœuds texte
 - Ses enfants sont ordonnés selon l'ordre de lecture du document

Nœuds

- Un **noeud attribut**
 - est étiqueté par le nom et la valeur de l'attribut qu'il représente
 - n'a pas de noeud fils
- Un **noeud texte**
 - est étiqueté par le fragment de texte qu'il représente
 - a pour père un noeud élément et n'a pas de noeud fils
- Un **noeud namespace**
 - Qualifier les noms d'attributs ou d'éléments intervenant dans certaines parties d'un document XML.
- Un **noeud commentaire**
 - Type d'un noeud commentaire XML <!-- -->

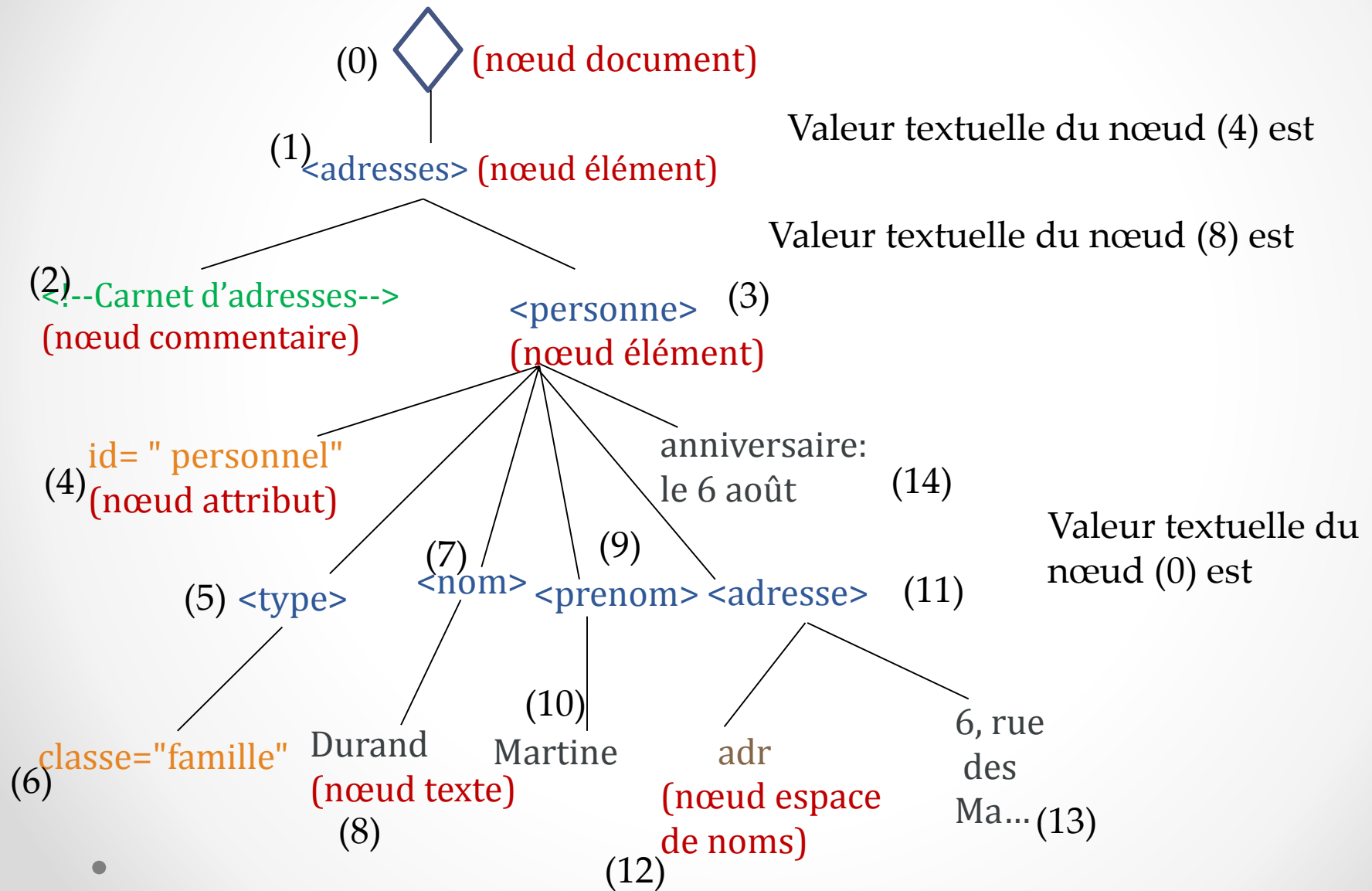
Ordre du document

Ordre du document = ordre de lecture de ses constituants représentés par des nœuds



Contraintes sur les nœuds

- Le **nœud document** peut avoir des nœuds fils qui peuvent être des nœuds commentaire, un nœud élément et instruction de traitement .
- Les **enfants** d'un nœud élément: nœuds éléments, texte, commentaire, instruction de traitement.
- Les attributs et espaces de noms n'ont pas de lien d'affiliation avec les éléments auxquels ils sont associés dans le document XML. Ils sont simplement liés à eux.
- Un nœud ne doit pas avoir deux enfants consécutifs qui sont des nœuds texte.
- Un nœud ne doit pas avoir des enfants qui sont des nœuds texte dont le contenu est vide.
- Les **descendants** du nœud racine ou d'un nœud élément sont ses enfants ou les enfants de ses enfants.
⇒ Les enfants et les descendants d'un nœud ne sont donc pas des nœuds de type attribut ou espace de noms.



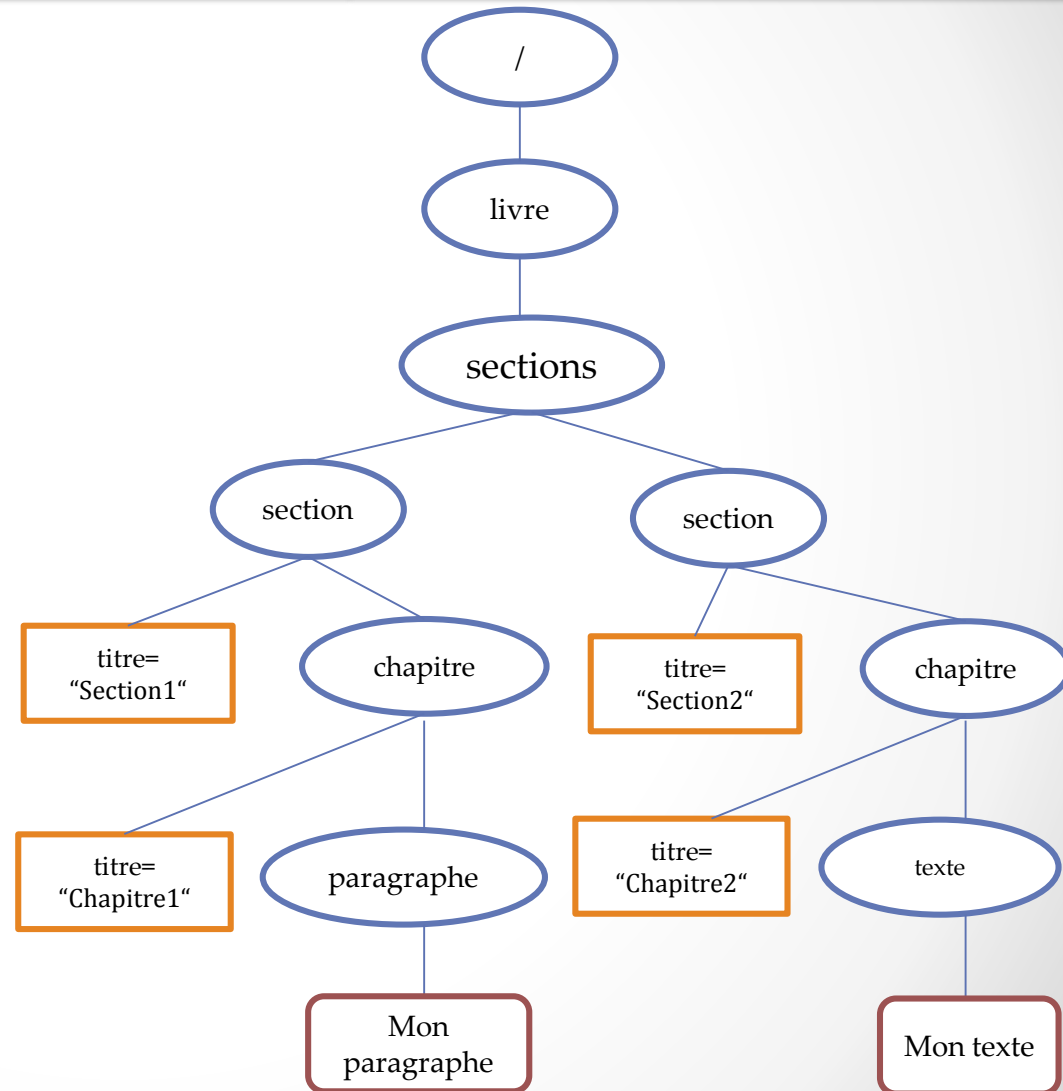
Pourquoi XPath?

- Localiser des fragments d'un document XML: des nœuds dans un arbre XML:
 - ➔ **Expressions de chemin** : abrégée ou basée sur des axes
- Une expression de chemins XPath permet de sélectionner des chemins à parcourir dans l'arbre du document en partant d'un nœud origine jusqu'à un ou plusieurs nœuds destination.

Introduction
Modèle de données XDM
Le langage XPATH

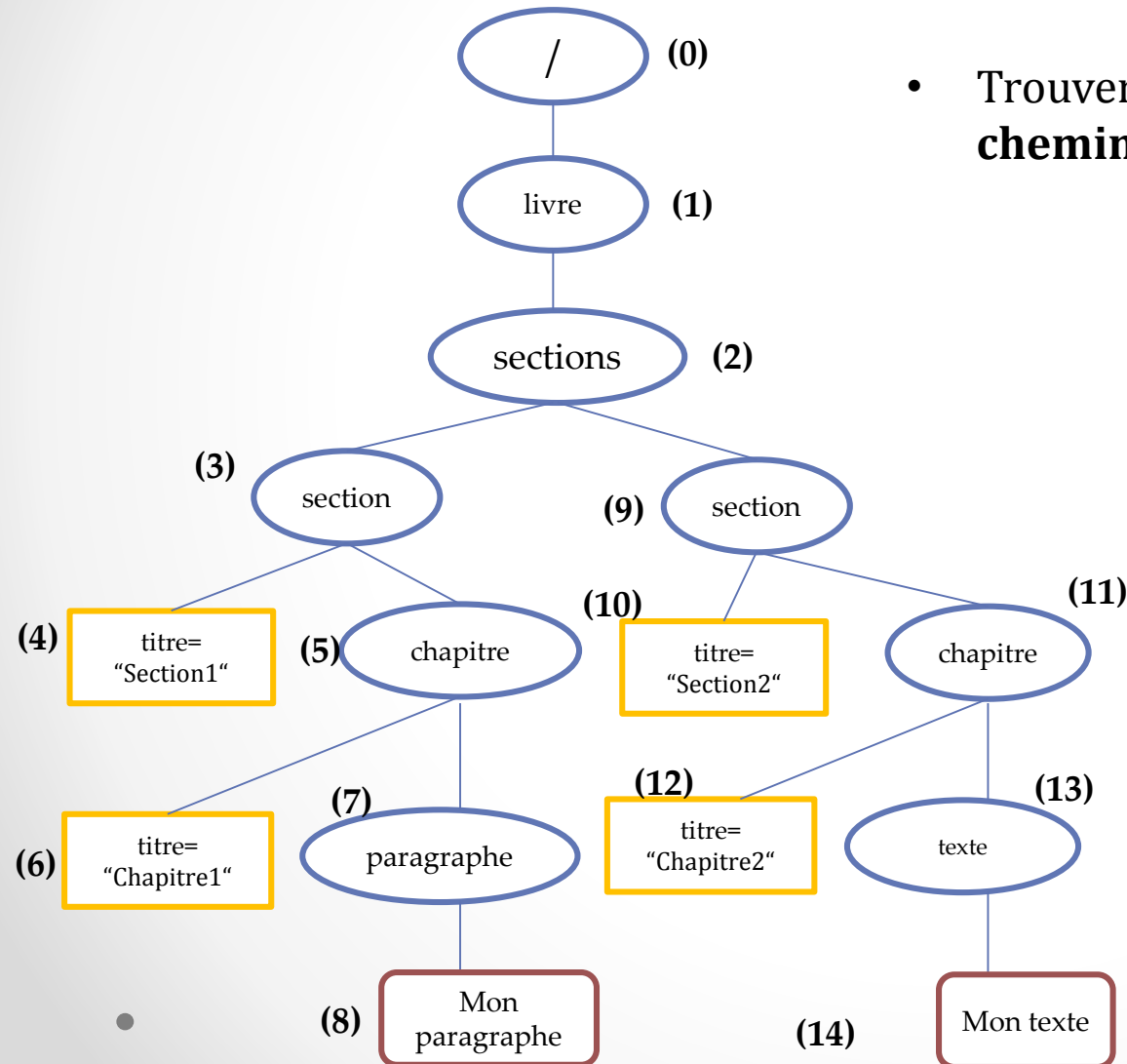
- Présentation
- Expressions XPath
- Axes de recherche
- Filtres
- Prédicats
- Simplification

```
<livre>
  <sections>
    <section titre="Section1">
      <chapitre titre="Chapitre1">
        <paragraphe>Mon paragraphe</paragraphe>
      </chapitre>
    </section>
    <section titre="Section2">
      <chapitre titre="Chapitre2">
        <texte>Mon texte</texte>
      </chapitre>
    </section>
  </sections>
</livre>
```



- Trouver la liste des nœuds sections: **Le chemin (0,1,2)** est conforme à ce modèle

/livre/sections



Expression de chemin

- Une expression de chemin XPath :
 - s'évalue en fonction d'un noeud **contexte** (qui est le nœud de départ),
 - désigne un ou plusieurs chemins dans l'arbre à partir du nœud contexte
- a pour résultat une instance du modèle XDM
 - un ensemble de noeuds (0 à plusieurs nœuds) ou
 - une valeur (numérique, booléenne ou alphanumérique)
 - une séquence (pour XPath 2.0)
- Consiste en une **séquence d'étapes** séparées par / ou //
- **L'opérateur // désigne les descendants**

Expressions relatives vs absolues

- Une expression (ou chemin) XPath peut être relative ou absolue (comme une URL)
 - *Expression **relative**:*
 - Comme A/B//C (A, B et C sont appelés **des sélecteurs**)
 - Relative au nœud courant (nœud quelconque du document).
 - *Expression **absolue**:*
 - Si elle commence par « / ». Exemple /A/B/C ou //A/B//C
 - Point de départ: le **nœud contexte** est le **nœud document**
 - Une expression absolue peut être considérée comme étant relative par rapport au nœud document.
- Chaque **sélecteur** sélectionne un ensemble de noeuds en fonction du résultat du sélecteur précédent.

Exemple d'évaluation: /livre/sections/section/@titre

- **1^{ère} étape:** /livre/sections/section/@titre

A partir de la **racine**, on cherche les noeuds de type élément de balise **livre**

- **2^{ème} étape:** sections/section/@titre

À partir d'un **noeud contexte** livre, on cherche les noeuds de type élément de balise **sections**

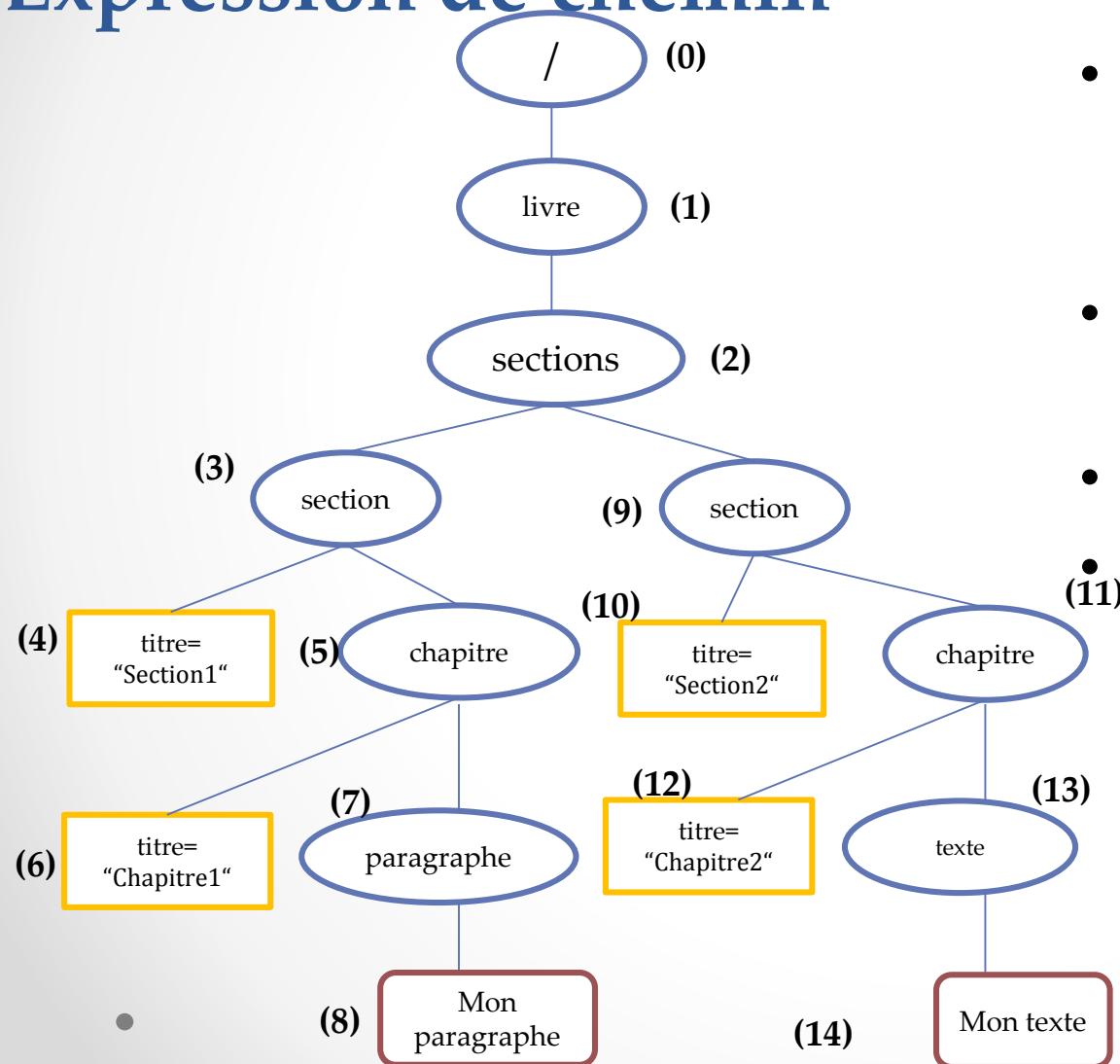
- **3^{ème} étape:** section/@titre

À partir d'un **noeud contexte** sections, on cherche les noeuds de type élément de balise **section**

- **4^{ème} étape:** @titre

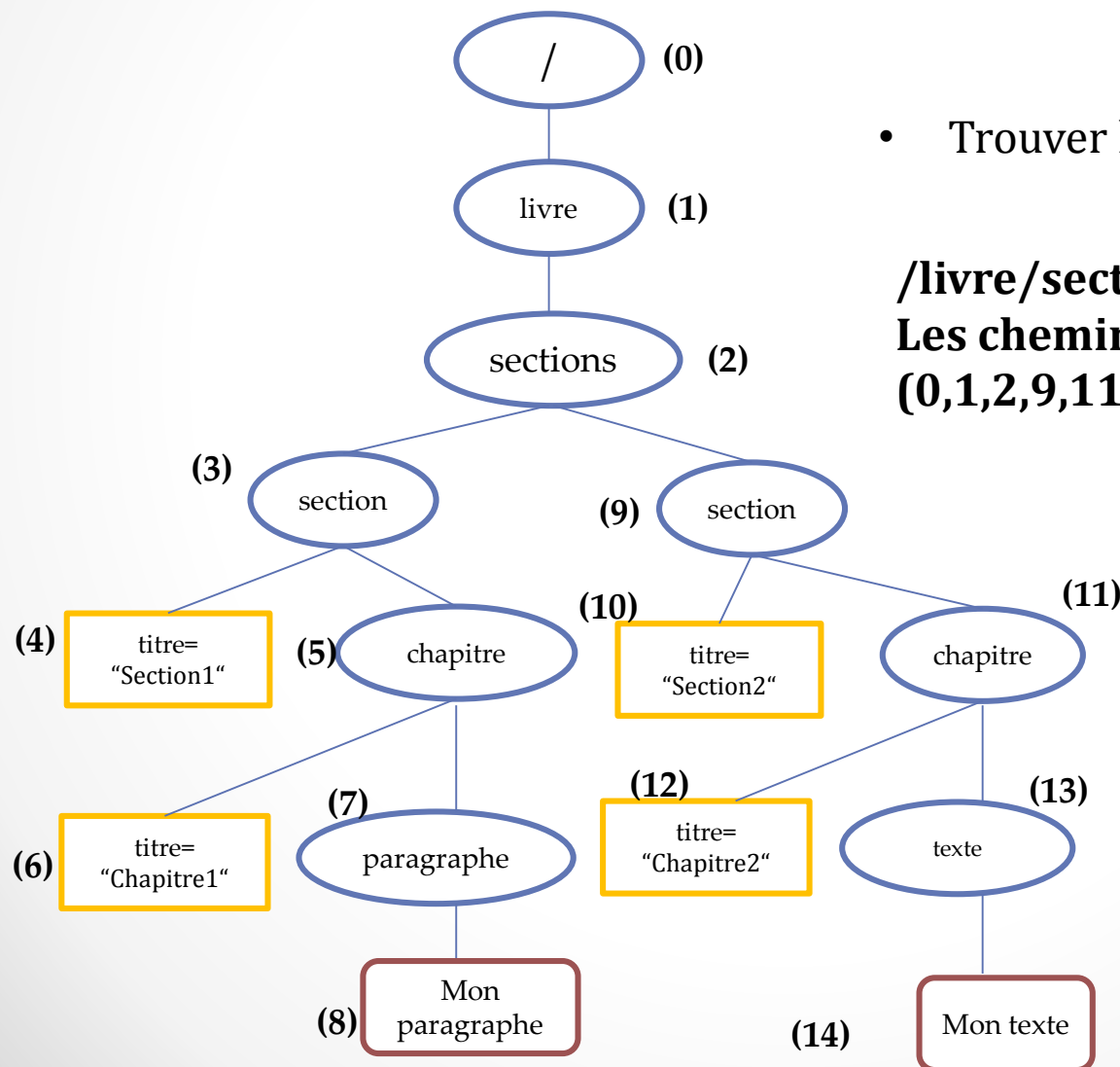
À partir d'un **noeud contexte** section on cherche les noeuds de type attribut **titre**.

Expression de chemin



- **1^{ère} étape:**
`/livre/sections/section/@titre`
- **2^{ème} étape:**
`sections/section/@titre`
- **3^{ème} étape:** `section/@titre`
- **4^{ème} étape:** `@titre`

Expression de chemin



- Trouver la liste des chapitres:

/livre/sections//chapitre

Les chemins (0,1,2,3,5) et

(0,1,2,9,11) sont conformes à ce modèle

À quoi correspondent les expressions suivantes:

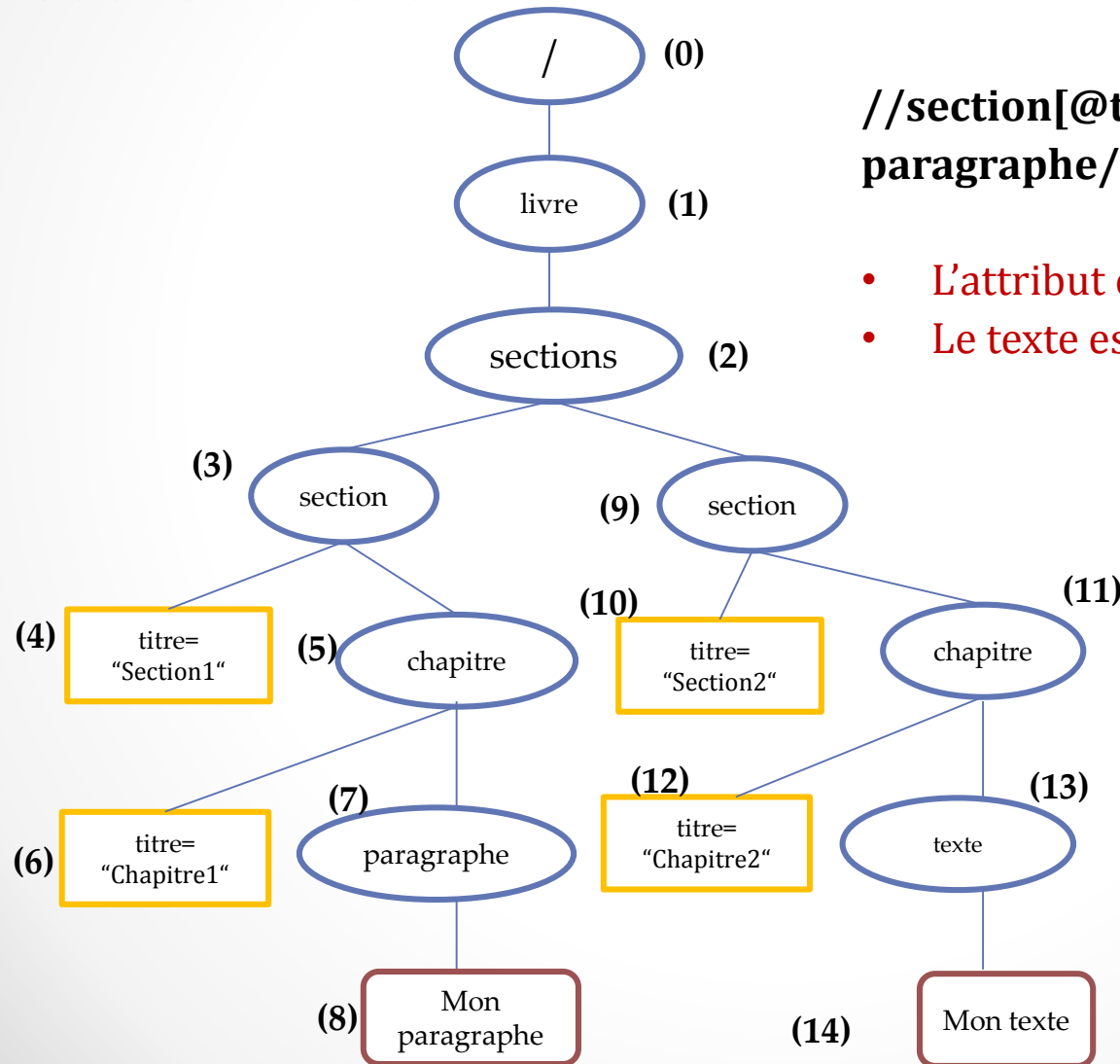
1) //auteur

2) //auteur/nom

3) /auteurs/personne/@nom

4) /auteurs/auteur/adresse/text()

Expression de chemin



**//section[@titre="Section1"]//
paragraphe/text()**

- L'attribut est préfixé par l'opérateur @
- Le texte est désigné par la fonction text().

Les sélecteurs de noeuds

- Les sélecteurs de nœuds sont de la forme: un axe, un filtre et une liste de prédicats en suivant la syntaxe:

axe:: filtre [predicat₁]...[predicat_n]

- **Un axe:** sens de navigation dans l'arbre par rapport au nœud contexte
 - **Filtre:** type de nœud à localiser
 - **Une suite de prédicats ou conditions(peut-être vide):** propriétés que doivent satisfaire les noeuds parmi les nœuds retenus
- Sa valeur est la séquence de nœuds atteints à partir du nœud contexte en suivant l'axe, vérifiant le filtre et les prédicats successifs

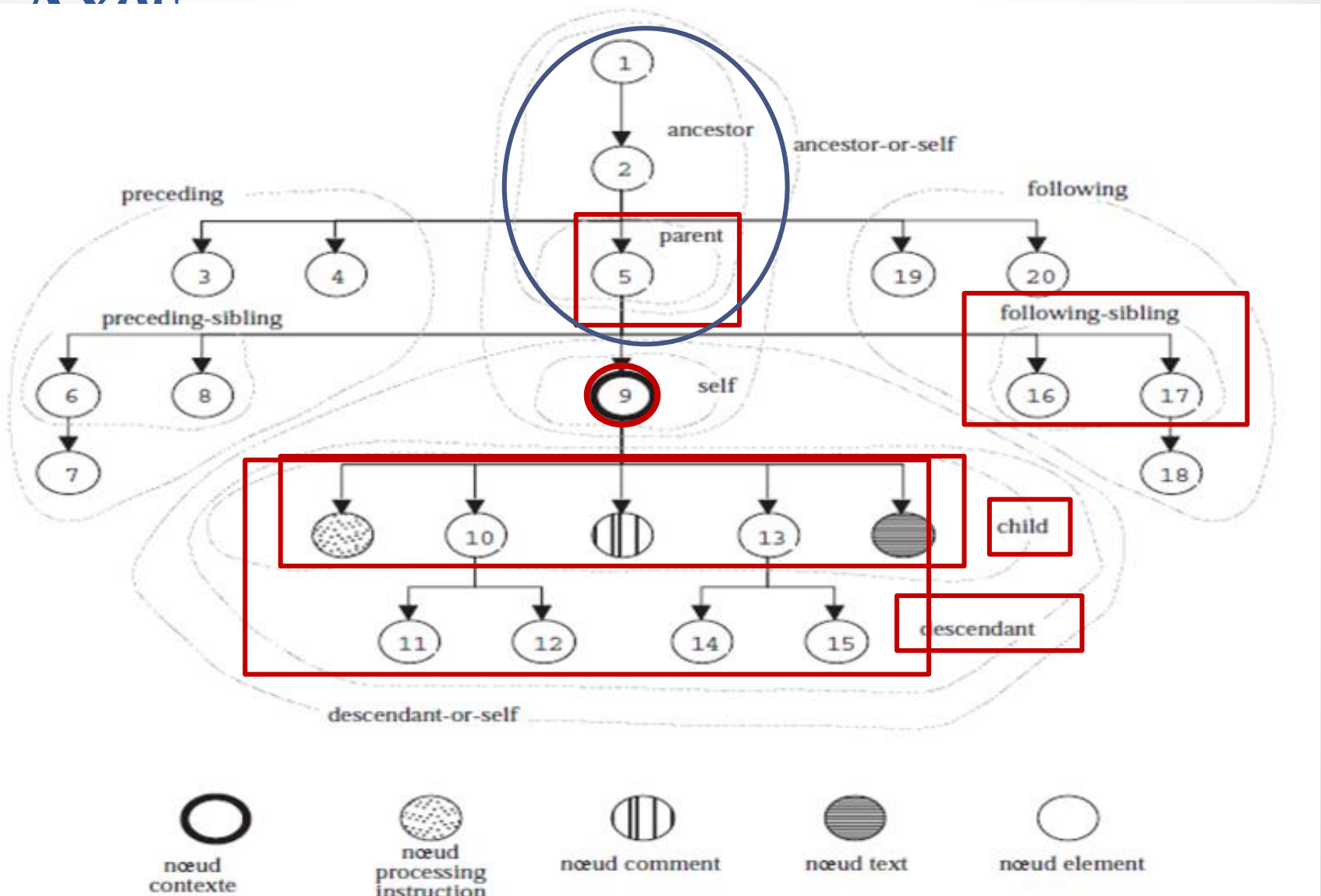
Les Axes

- Un axe sélectionne, dans un arbre et à partir du noeud courant, l'ensemble des noeuds qui peuvent être atteints en suivant une certaine direction.
- Un axe a un **sens** : **avant** ou **arrière**.
 - Sens avant:
Child, Descendant, descendant-or-self, Following, following-sibling
 - Sens arrière:
Parent, ancestor, ancestor-or-self, Preceding-sibling, preceding.
- Sans oublier
 - l'axe **attribute** pour les attributs de l'élément courant ;
 - **self** pour le nœud courant
 - **namespace** qui sélectionne les nœuds de type espace de nommage attachés au nœud courant.

Introduction
Modèle de données XDM
Le langage XPATH

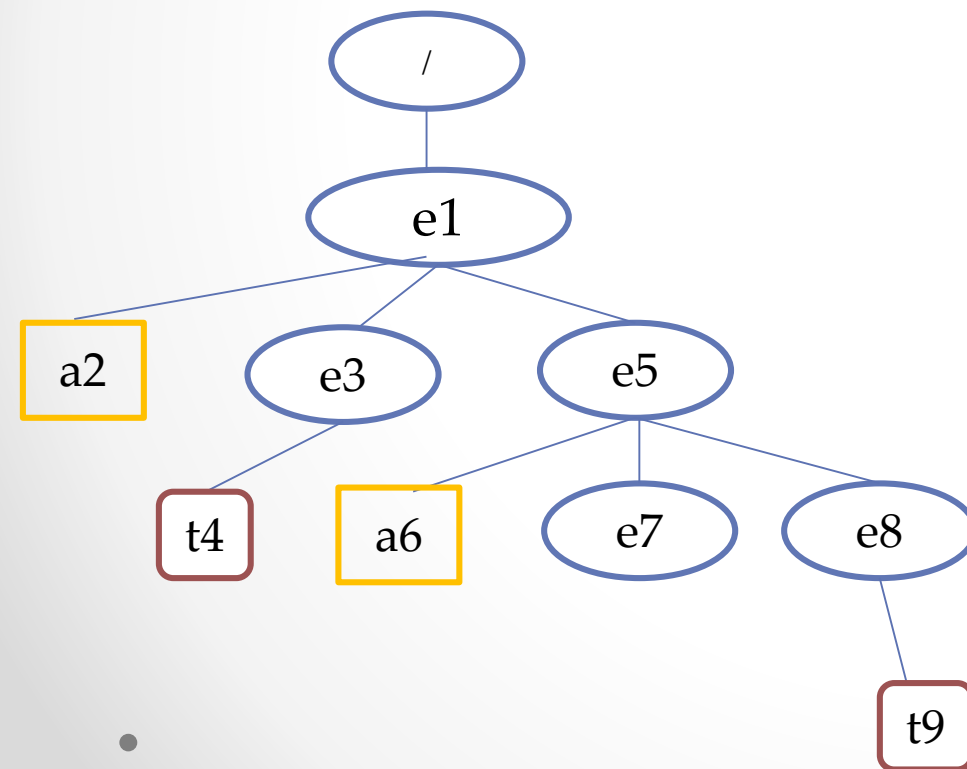
- Présentation
- Expressions XPath
- Axes de recherche
- Filtres
- Prédicats
- Simplification

Les Axes



Axe : Self

Self: sélectionne le nœud contexte (ou courant) et seulement le nœud contexte



- **Expression:** self::*
- **Sens** = avant ou arrière
- **Sorte de nœud principal:** élément
- **Noeud contexte** = e1
- **Noeuds sélectionnés** (dans l'ordre) = e1

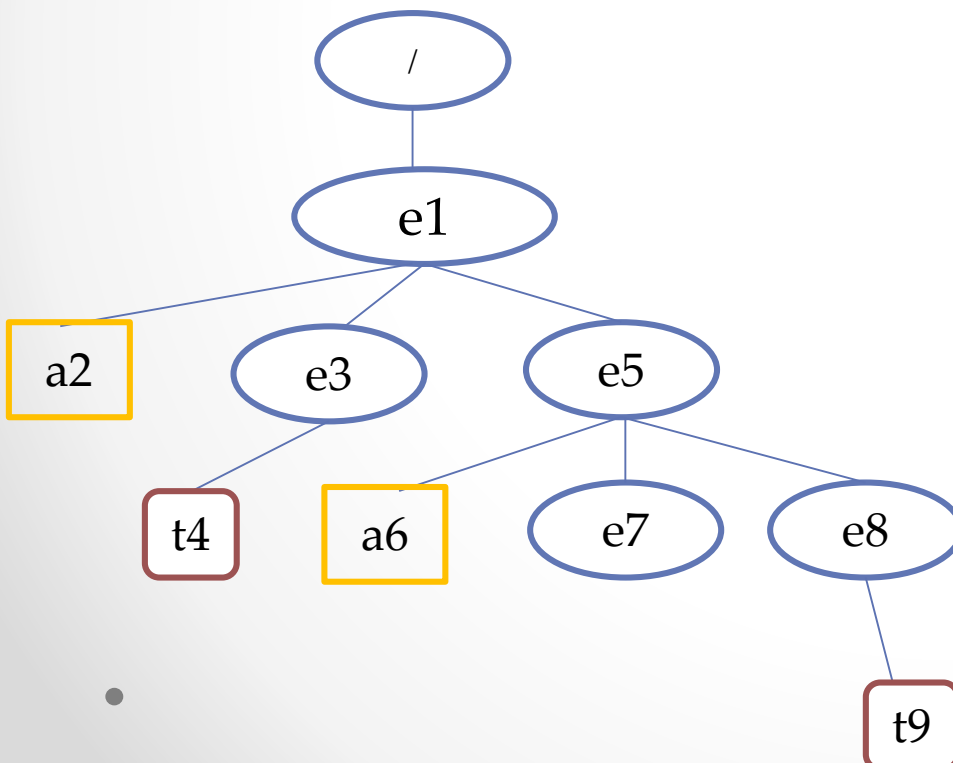
e: élément (ovale)

a: attribut (rectangle)

t: texte (rectangle à coins arrondis)

Axe : Child

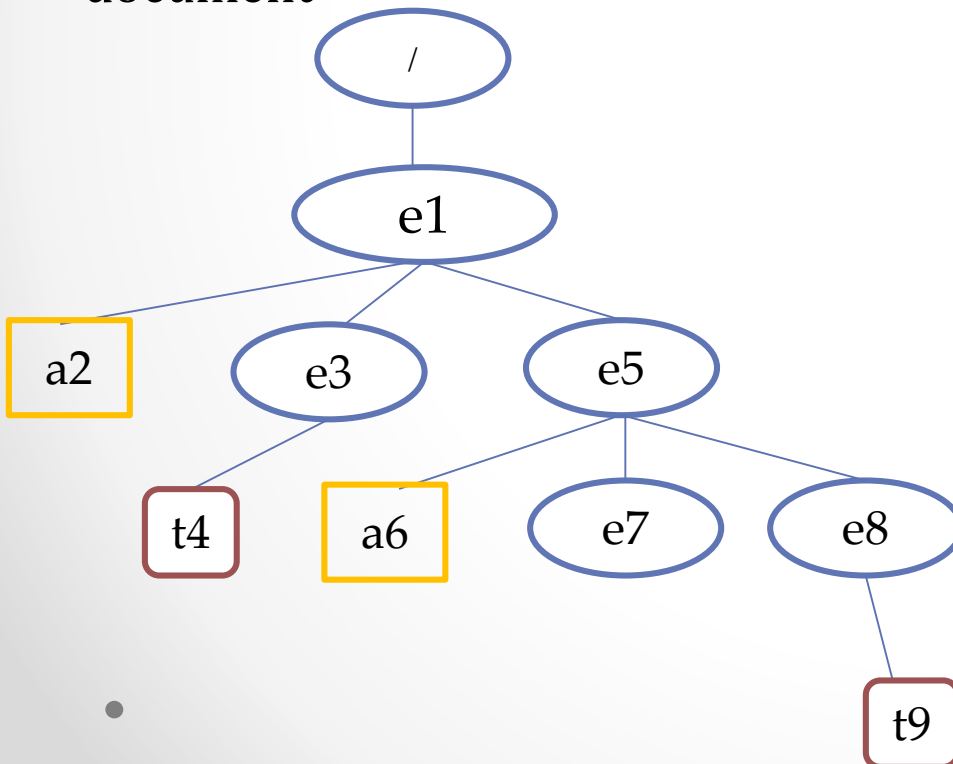
Child: sélectionne les fils directs du nœud courant (ne contient jamais de nœud de type attribut) dans l'ordre d'apparition dans le document



- **Expression:** `child::*`
- **Sens** = avant
- **Sorte de nœud principal:** élément
- **Noeud contexte** = e1
- **Noeuds sélectionnés** (dans l'ordre) = e3, e5

Axe : Descendant

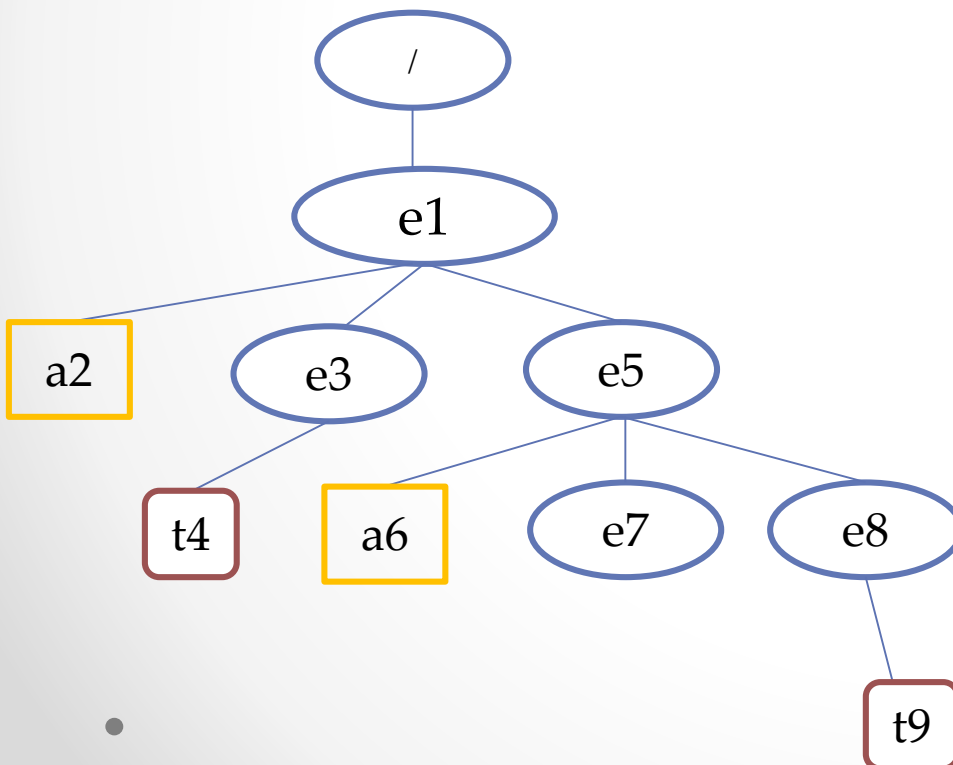
Descendant: sélectionne les descendants du noeud courant (ne contient jamais de nœud de type attribut) dans l'ordre d'apparition dans le document



- **Expression:** descendant::text()
- **Sens** = avant
- **Sorte de nœud principal:** élément
- **Noeud contexte** = /
- **Noeuds sélectionnés** (dans l'ordre) = t4,t9

Axe : descendant-or-self

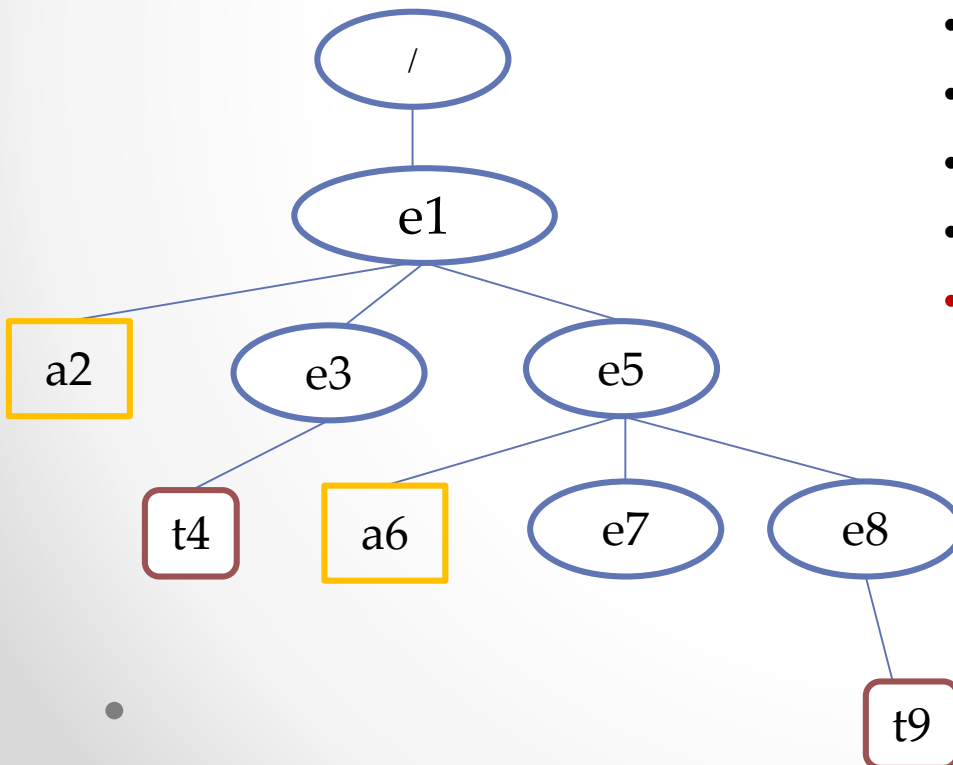
descendant-or-self: sélectionne le nœud contexte et tous ses descendants dans l'ordre d'apparition dans le document



- **Expression:** descendant-or-self::*
- **Sens** = avant
- **Sorte de nœud principal:** élément
- **Noeud contexte** = e5
- **Noeuds sélectionnés** (dans l'ordre) = e5, e7, e8, t9

Axe : parent

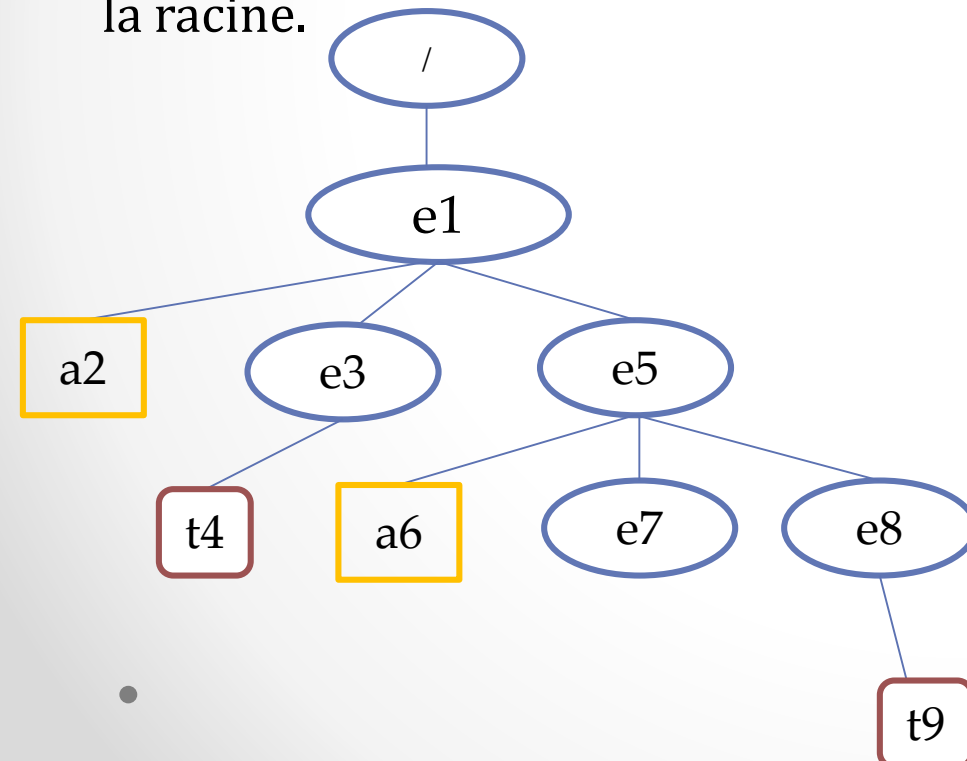
parent : sélectionne le noeud père du noeud courant, s'il en existe un



- **Expression:** parent::*
- **Sens** = arrière
- **Sorte de nœud principal:** élément
- **Noeud contexte** = e5
- **Noeuds sélectionnés** (dans l'ordre) = e1

Axe : ancestor

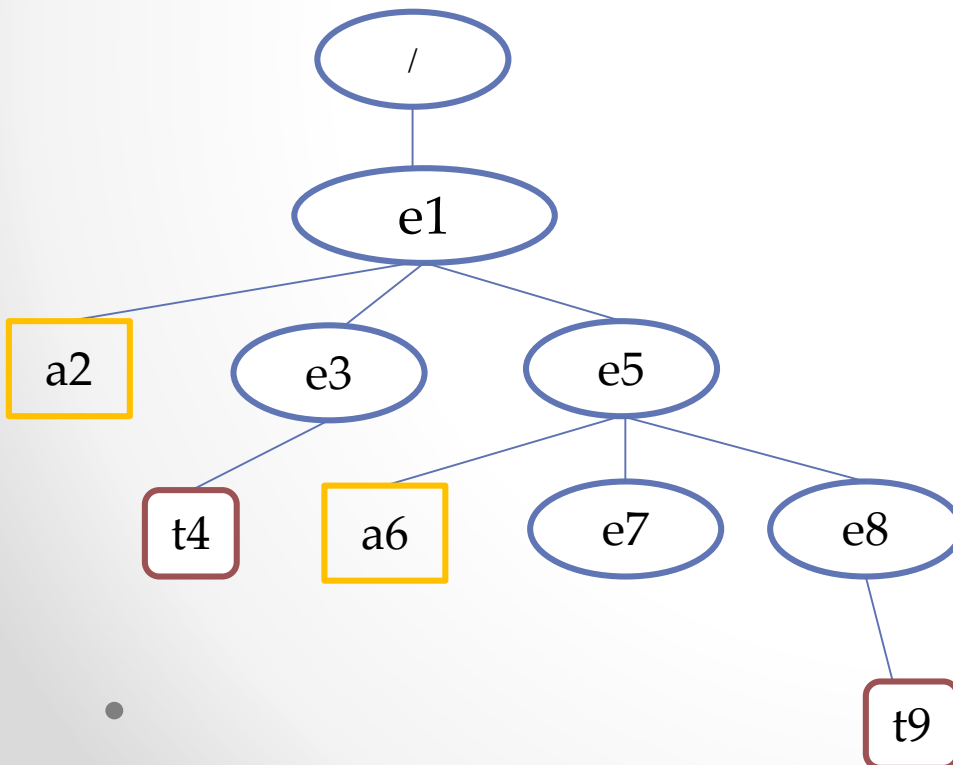
ancestor : sélectionne les ancêtres (parents, grand-parent, ...) du nœud contexte dans l'ordre inverse d'apparition dans le document. Cet axe contient toujours le nœud racine, sauf si le nœud contextuel est lui-même la racine.



- **Expression:** ancestor::*
- **Sens** = arrière
- **Sorte de nœud principal:** élément
- **Noeud contexte** = e7
- **Noeuds sélectionnés** (dans l'ordre) = e5, e1, /

Axe : ancestor-or-self

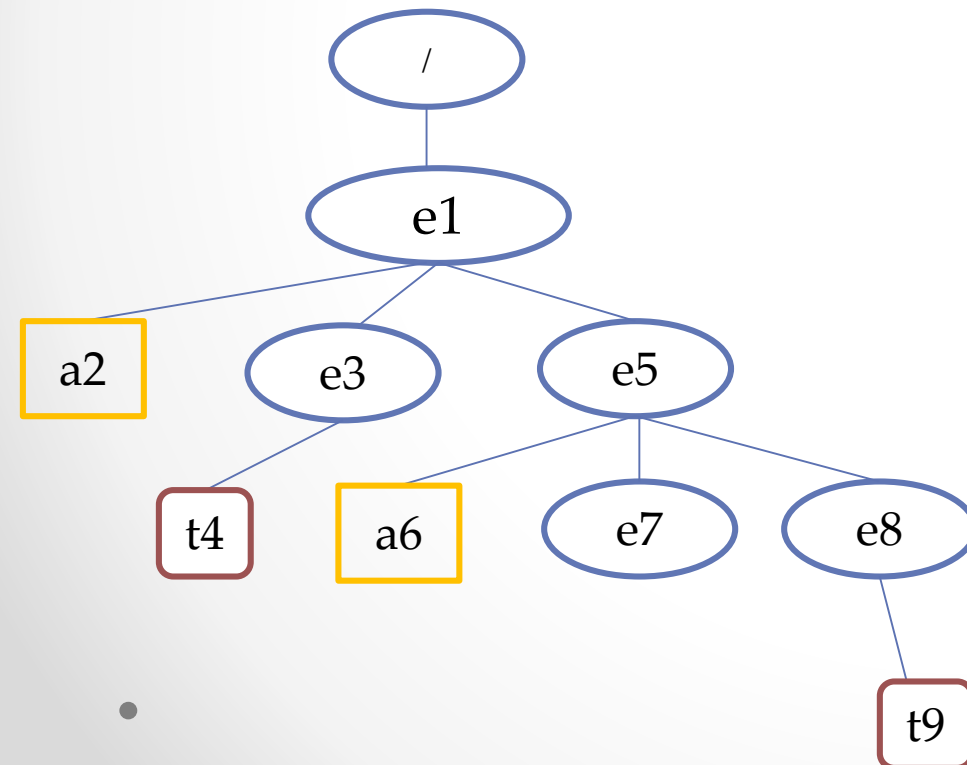
ancestor-or-self: sélectionne le nœud contexte et tous ses ancêtres dans l'ordre inverse d'apparition dans le document



- **Expression:** ancestor-or-self::*
- **Sens** = arrière
- **Sorte de nœud principal:** élément
- **Noeud contexte** = t9
- **Noeuds sélectionnés** (dans l'ordre) = t9, e8, e5, e1, /

Axe : following-sibling

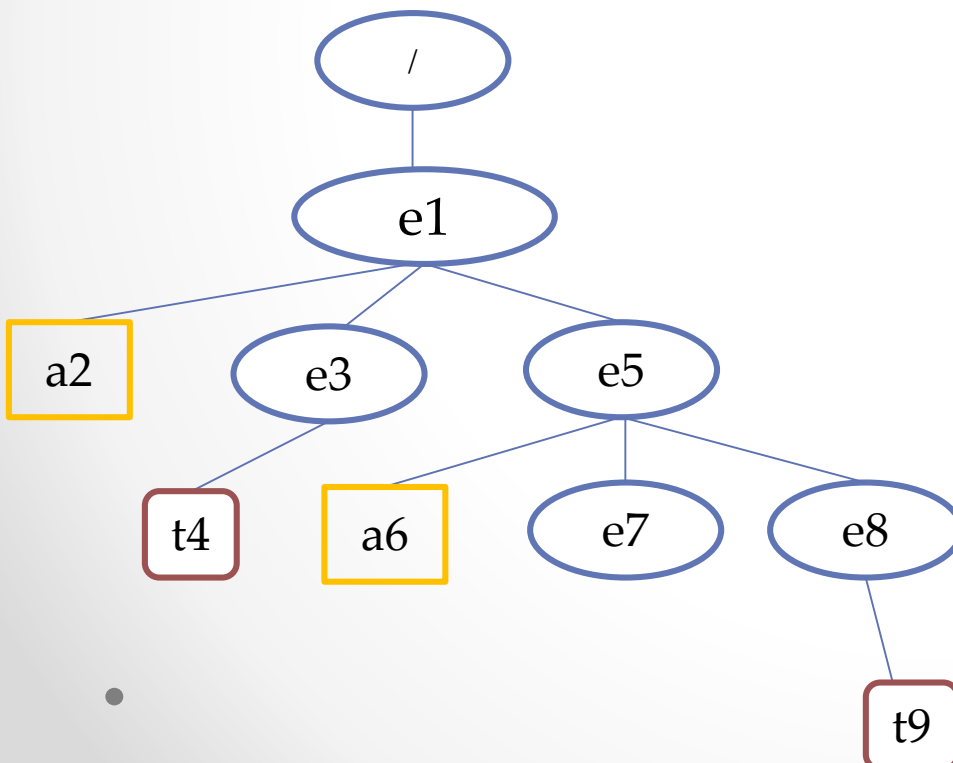
following-sibling: sélectionne les éléments frères droits (suivants) du nœud courant dans l'ordre d'apparition dans le document



- **Expression:** following-sibling::*
- **Sens** = avant
- **Sorte de nœud principal:** élément
- **Noeud contexte** = e3
- **Noeuds sélectionnés (dans l'ordre)** = e5

Axe : preceding-sibling

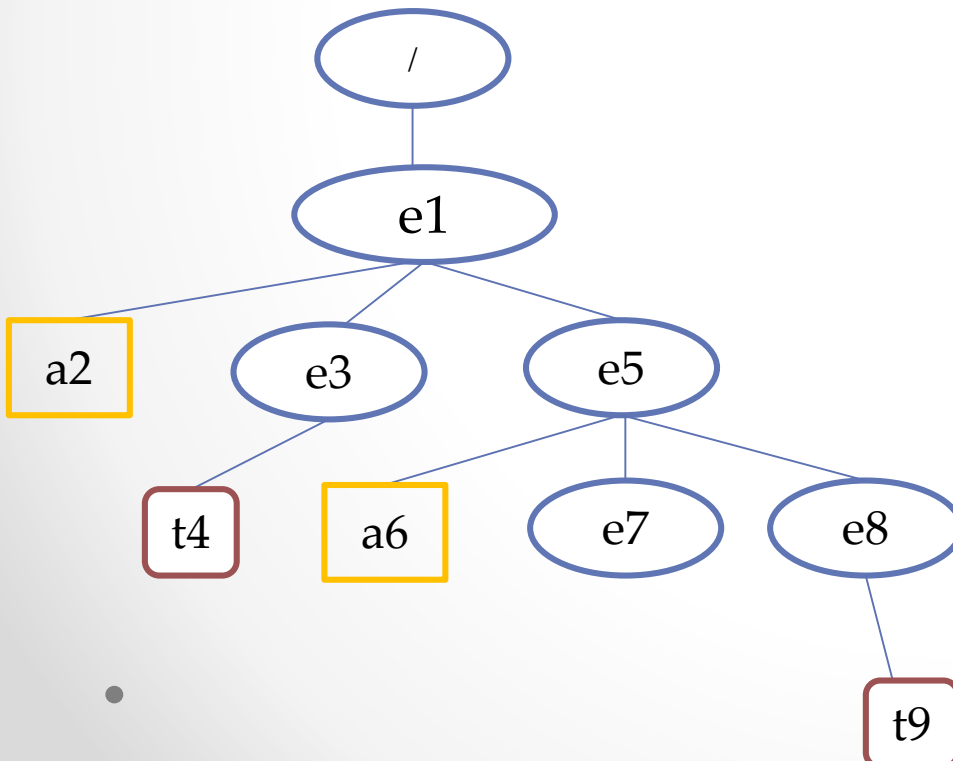
preceding-sibling: sélectionne les éléments frères gauches du nœud courant dans l'ordre inverse d'apparition dans le document



- **Expression:** Preceding-sibling::*
- **Sens** = arrière
- **Sorte de nœud principal:** élément
- **Noeud contexte** = e5
- **Noeuds sélectionnés** (dans l'ordre) = e3

Axe : following

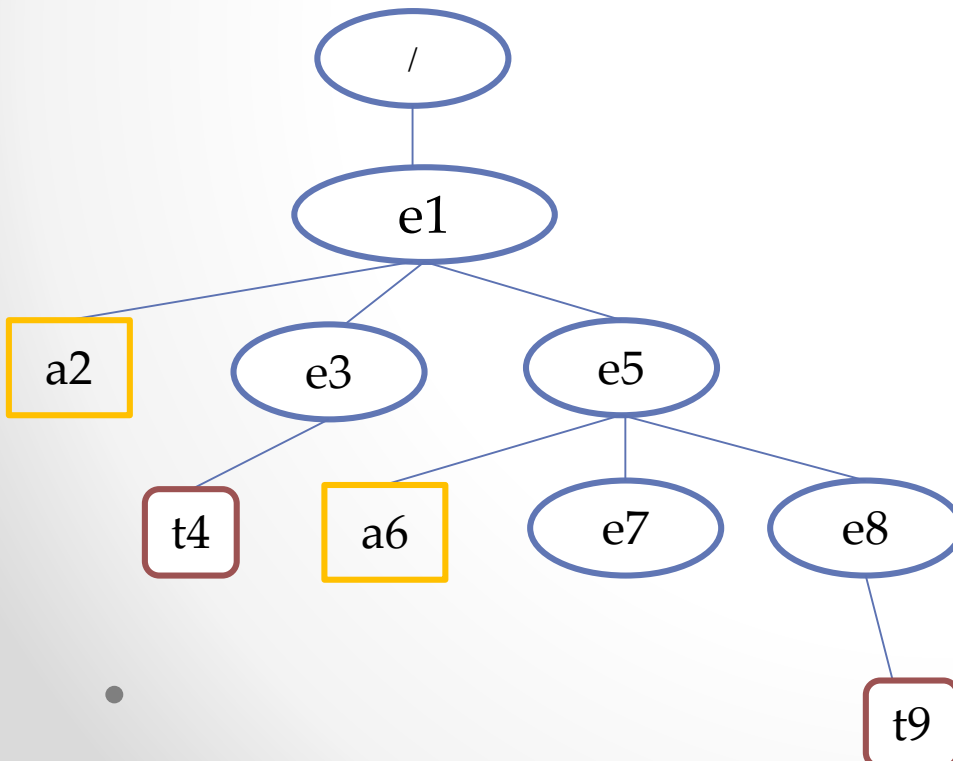
Following: sélectionne les nœuds suivants le nœud courant dans l'ordre d'apparition dans le document excepté les descendants du nœud courant ainsi que tous les nœuds attributs et espace de noms qui lui sont associés



- **Expression:** Following::*
- **Sens** = avant
- **Sorte de nœud principal:** élément
- **Nœud contexte** = e7
- **Nœuds sélectionnés** (dans l'ordre) = e8, t9

Axe : Preceding

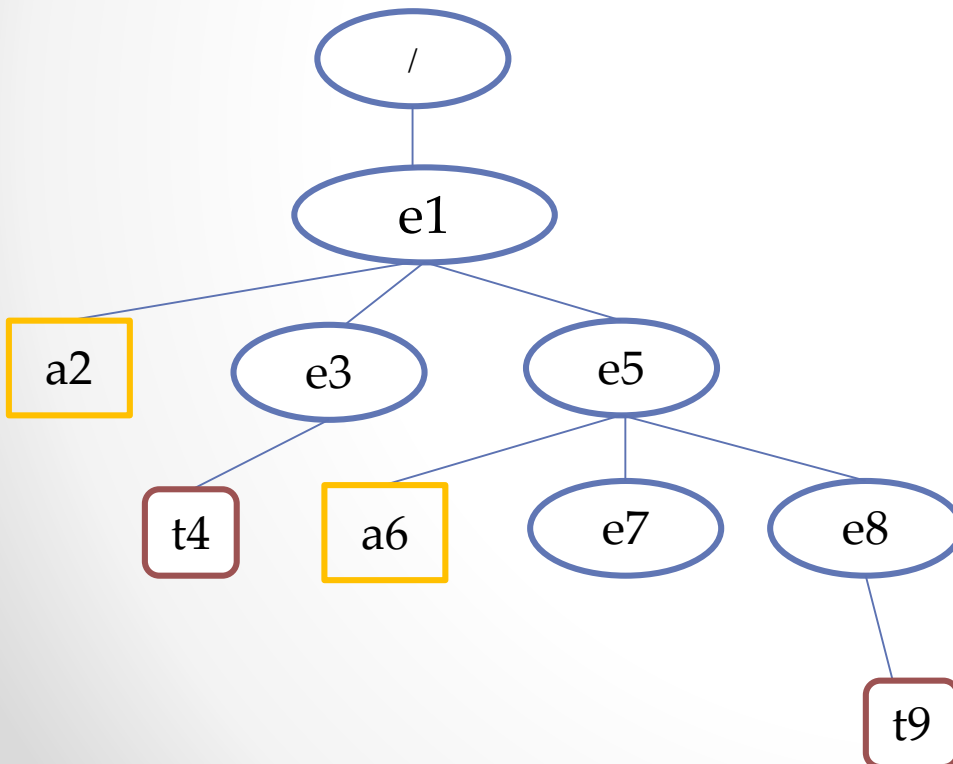
Preceding: sélectionne les nœuds précédant le nœud courant dans l'ordre inverse du document excepté les ancêtres du nœud courant ainsi que les nœuds attributs et espace de noms qui lui sont associés



- **Expression:** Preceding::*
- **Sens** = arrière
- **Sorte de nœud principal:** élément
- **Nœud contexte** = e8
- **Nœuds sélectionnés** (dans l'ordre) = e7,t4,e3

Axe : attribute

attribute: sélectionne les nœuds de type attribut liés au nœud courant



- **Expression:** Attribute::*
- **Sens** = avant
- **Sorte de nœud principal:** attribut
- **Noeud contexte** = e5
- **Noeuds sélectionnés** (dans l'ordre) = a6

Filtre(ou test de nœud)

- Un filtre permet de sélectionner parmi les nœuds de l'axe ceux qui sont d'un certain type. Un filtre peut être:
 - **un nom** d'élément ou d'attribut

`/livre/auteurs/auteur/attribute::nom`
 - **le caractère *** qui sélectionne tous les nœuds qui ont un nom
 - `child::*` sélectionne tous les éléments fils du nœud courant,
 - `attribute::*` sélectionne tous les attributs du nœud courant.

Filtre(ou test de nœud)

- **text()** qui sélectionne les nœuds de type texte

```
//section/paragraphe/text()
```

- **comment()** qui sélectionne les nœuds de type commentaire

- **node()** qui sélectionne tous les nœuds.

```
/livre/sections/section/chapitre/node()
```

- **processing-instruction()** qui sélectionne tous les nœuds de type instruction de traitement de l'axe

Prédicat ou condition

- Un prédicat est une **expression booléenne** (placée entre **crochets**) qui peut être évaluée à vrai ou faux, construite à partir :
 - d'expressions de chemin et/ou
 - de fonctions prédéfinies:
 - Fonctions sur les nombres
 - Fonctions sur les booléens
 - Fonctions sur les nœuds
 - Fonctions sur les chaînes
 - Ainsi que des opérations et des conditions logiques

Condition d'existence

- `//child::element[n]`
 - Sélectionne le nième élément *element* dans le nœud courant.
- `//child::element[elt]`
 - Sélectionne dans le nœud courant, l'élément *element* qui a comme élément fils *elt*.
- `//child::element[@attribut]`
 - Sélectionne dans le nœud courant, l'élément *element* qui possède un attribut *attribut*
- `// child::element[@attribut="valeur"]`
 - Sélectionne dans le nœud courant, l'élément dont l'attribut *attribut* a une valeur égale à *valeur*

Fonctions

Fonctions sur les nombres

Fonction	Définition
sum(<i>noeuds</i>)	renvoie la somme des noeuds après les avoir transformés en nombre.
count(<i>noeuds</i>)	renvoie le nombre de nœuds.
floor(<i>nb</i>)	retourne le plus grand nombre entier plus petit ou égal à nb
round(<i>nb</i>)	arrondi par le plus proche

Fonctions sur les booléens

Fonction	Définition
true()	Toujours vraie
false()	Toujours fausse
not(booléen)	Vraie ssi le paramètre est faux

Fonctions sur les nœuds

Fonction	Définition
last()	vraie ssi le nœud est le dernier du contexte courant
position ()	retourne le numéro d'ordre du nœud dans le nœud contexte courant. La première position vaut 1, la dernière vaut last ().
name(nœuds)	Renvoie l'étiquette d'un nœud
id(nom)	Renvoie le nœud identifié par l'étiquette nom

Fonctions sur les chaînes

Fonction	Définition
string(object)	Convertit l'objet donné en argument en chaîne de caractères
contains (ch1,ch2)	teste si chaîne1 contient chaîne2
Concat(ch1,.....,chn)	Concaténation de chaînes

Et bien d'autres....

Opérations et conditions logiques

- Pour affiner des requêtes XPath, il est possible d'utiliser des opérateurs dans un prédicat:

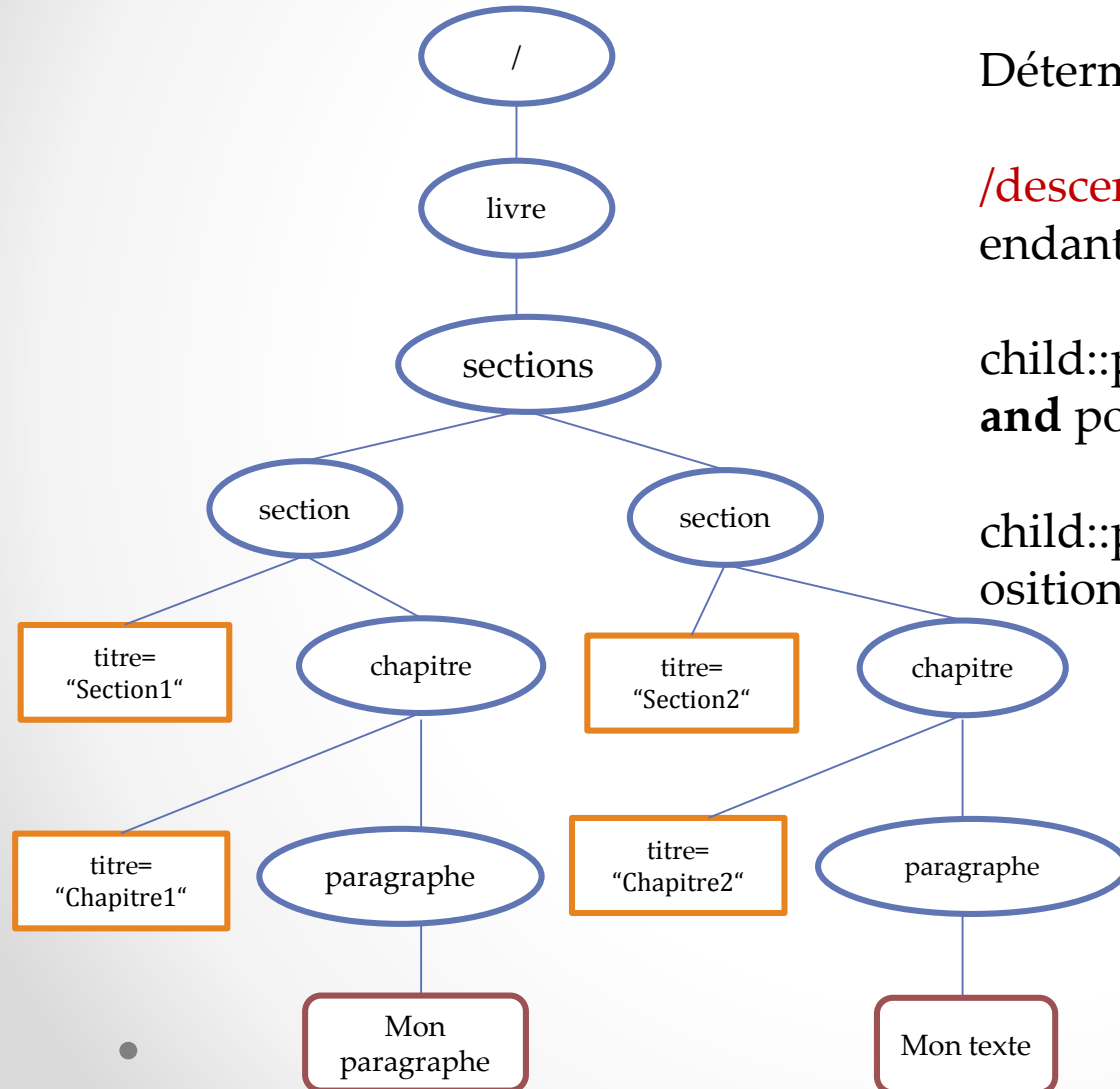
Opérateur XPath	Définition
=	Égalité
!=	Différence
> , < , >= , <=	Supérieur, inférieur, supérieur ou égal, inférieur ou égal
and	« ET » logique
or	« ou » logique
Not()	Négation logique
+ , -	Addition, soustraction
Div	Division
Mod	Reste de la division entière (modulo)

- Chapitre[@id>3]: sélectionne tous les nœuds chapitre du nœud courant ayant leur id supérieur à 3

- `child::paragraphe[child::figure]`
- `child::paragraphe[child::*]`
- `child::*[self::chapitre or self::annexe]`
- `child::paragraphe[child::figure[position() = 2]]`

Exercice

- Trouver la liste des éléments ayant un seul attribut
- Trouver la liste des chapitres contenant deux paragraphes



Déterminer le résultat de ces expressions:

/descendant::section[2]/child::chapitre/descendant::text()

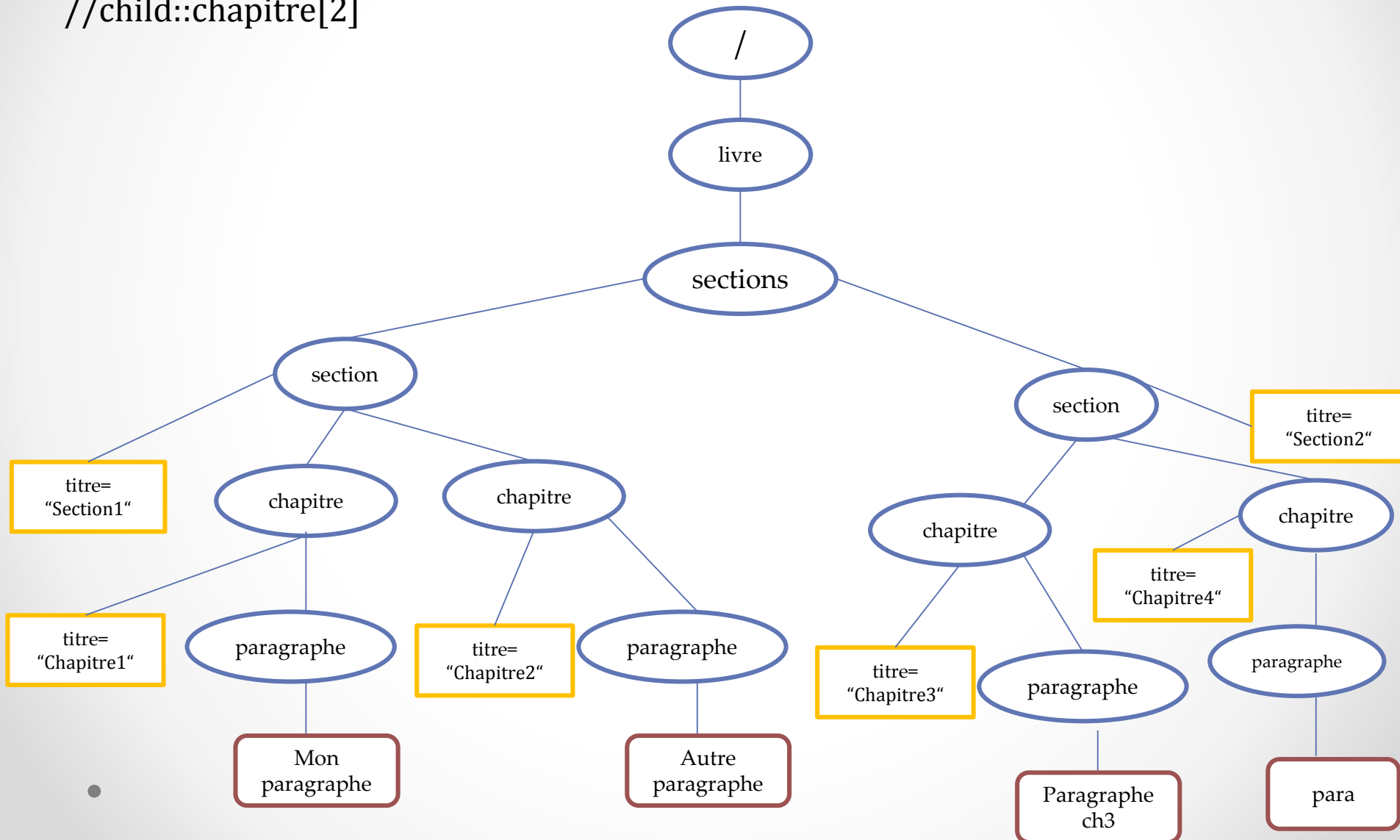
child::paragraphe[contains(text(),'Mon') and position()=2]

child::paragraphe[contains(text(),'Mon')][position()=2]

Introduction
Modèle de données XDM
Le langage XPATH

- Présentation
- Expressions XPath
- Axes de recherche
- Filtres
- Prédicats
- Simplification

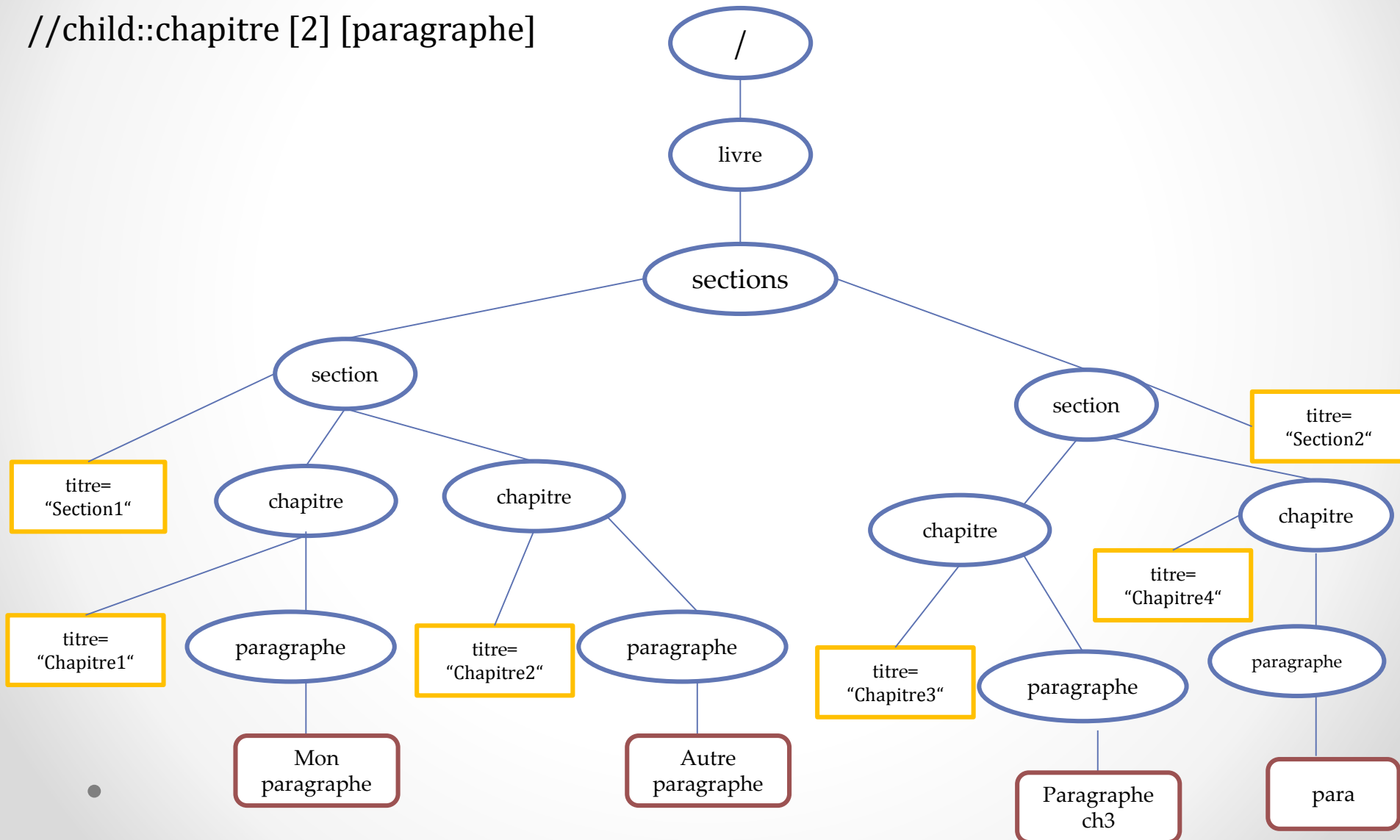
//child::chapitre[2]



Introduction
Modèle de données XDM
Le langage XPATH

- Présentation
- Expressions XPath
- Axes de recherche
- Filtres
- Prédicats
- Simplification

//child::chapitre [2] [paragraphe]



Évaluation d'une expression de chemin

- `axe::filtre [exp1]...[expn]`
- A partir de S, une séquence de noeuds
- Pour chaque noeud de S (**noeud contexte**)
 - On calcule la séquence de noeuds N sélectionnés par l'axe puis le filtre
 - On calcule la sous séquence N1 de N vérifiant exp1
 - calcule la sous séquence N2 de N1 vérifiant exp2
 - ...
 - On calcule la sous séquence Nn de Nn-1 vérifiant expn

Résultat

- l'union des séquences Nn des noeuds atteints à partir de chaque noeud de S

Syntaxe abrégée

Syntaxe abrégée	Syntaxe étendue
filtre	child::filtre
@filtre	attribute::filtre
//filtre	/descendant-or-self:: node()/child::filtre
.	self::node() (se sélectionner soit même)
..	parent::node()
[x]	[position()=x]
./filtre	descendant-or-self:: node()/child::filtre

Exercice

Utiliser les axes pour exprimer les requêtes suivantes:

1) `./para`

2) `../title`

Exercice

```
<livre titre="Mon livre">
  <auteurs>
    <auteur nom="nom1" prenom="prenom1"/>
    <auteur nom="nom2" prenom="prenom2"/>
  </auteurs>
  <sections>
    <section titre="Section1">
      <chapitre titre="Chapitre1">
        <paragraphe>Premier paragraphe</paragraphe>
        <paragraphe>Deuxième paragraphe</paragraphe>
      </chapitre>
    </section>
    <section titre="Section2">
      <chapitre titre="Chapitre1">
        <paragraphe>Premier paragraphe</paragraphe>
        <paragraphe>Deuxième paragraphe</paragraphe>
      </chapitre>
    </section>
  </sections>
</livre>
```

En utilisant les axes, et forme abrégée (pour les deux premières), déterminer:

- La liste des attributs titre
- La liste des chapitres de la première section
- La liste des éléments ayant un ancêtre sections

Liens utiles

- Modélisation XDM <http://www.w3.org/TR/xpath-datamodel/>
- XPath recommandation W3C: <http://www.w3.org/TR/xpath20/>
- <http://xmlfr.org/w3c/TR/xpath/>
- <http://msdn.microsoft.com/fr-fr/library/ms256115.aspx>
- Support de Cours en ligne Elisabeth Murisasco
- Jacques Le Maitre, Description et manipulation de documents XML, supports de cours en ligne <http://lemaitre.univ-tln.fr/cours.htm>