



Initiation à XML



eXtensible Markup Language

Antonin Benoît DIOUF

Plan

Les composants d'un document XML

Les DTD et les schémas

Les feuilles de style

Les entités

Les espaces de noms

La galaxie « xml »

Composants

- ❑ **Le prologue** : ligne d'introduction d'un document XML (obligatoire). Il permet d'indiquer la version XML utilisée, le jeu de caractères (format d'encodage de caractères) utilisé et l'autonomie du document.
- ❑ **L'élément racine** : Créé par soi-même il est unique dans le document XML et encadre son contenu (obligatoire aussi)
- ❑ **Le contenu** : succession d'éléments et de sous-éléments contenus par l'élément racine.



Il est impératif de respecter la casse dans un document XML, par exemple `<livre>` et `<Livre>` ne signifient pas la même chose.

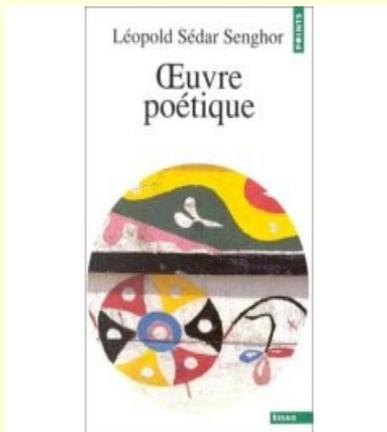
L'élément XML

- un élément comporte une balise ouvrante et une balise fermante
- il peut contenir
 - une chaîne de caractères `<element> texte </element>`
 - un ou plusieurs autres éléments
`<element> <sous_element> texte </sous_element> </element>`
 - une chaîne de caractères et un ou plusieurs éléments `<element> texte1 <sous_element> texte2 </sous_element> </element>`



la chaîne de caractères ne doit pas comporter les signes `<`, `>`, `&`, `"`, et `'` qui ont une signification particulière en XML ; pour les afficher, on utilise des appels d'entités caractères (voir plus loin)

Exemple de document XML



Titre : Œuvre poétique

Auteur : Senghor, Léopold Sedar (1906-2001)

Publication : Paris : Ed. du Seuil, 1990

Description matérielle : 429 p. ; 22 cm

ISBN : 2-02-012106-9

Prix : 38,11 EUR

```
<?xml version="1.0" encoding="UTF-8" ?>
```

le « prologue »

```
<livre>
```

```
<titre>Œuvre poétique </titre>
```

```
<auteur>Senghor, Léopold Sedar (1906-2001)</auteur>
```

```
<edition>Paris : Ed. du Seuil, 1990</edition>
```

```
<description>429 p. ; 22 cm</description>
```

```
<ISBN>2-02-012106-9</ISBN>
```

```
<prix>38,11 EUR</prix>
```

```
</livre>
```

l'élément racine

L'attribut XML

- les attributs ajoutent une information sur la valeur de l'élément

- Exemples :

```
<titre>Murambi</titre>
```

```
<titre type="complement">Le livre des ossements</titre>
```

```
<titre xml:lang="fr" >Une si longue lettre</titre>
```

```
<titre xml:lang="en" >So long a letter</titre>
```

L'attribut XML

4 attributs normalisés par le W3C

xml:lang précise la langue de la valeur de l'élément

xml:space permet de définir le mode de traitement des espaces blancs (espaces, tabulations, interlignes)

xml:base permet de définir une URL qui sert de répertoire de base au niveau du document; à utiliser en relation avec `xlink:href`

xml:id permet de donner à l'élément un identifiant unique au sein d'un document ou d'un ensemble de documents

Les règles de construction

(rappel)

- Tout document XML commence par le « prologue »
- Il faut spécifier le format d'encodage des caractères
- Les documents XML ont toujours un et un seul élément racine
- Les noms des éléments sont sensibles à la casse
- Les éléments doivent toujours être fermés
- Les éléments doivent toujours être correctement emboîtés (pas de chevauchement)

`<adresse>10<avenue>avenue Lamine Guèye</adresse></avenue>`

- Les attributs d'éléments doivent toujours être entre guillemets
- Il y a seulement cinq entités définies par défaut (<, >, &, ", et ')

=> si un document respecte l'ensemble de ces règles, il est « **bien formé** » et peut être analysé par un **parser** XML

Document XML bien formé

Pour s'en assurer visuellement, il faut utiliser un

Parser : analyseur syntaxique (parseur en français)

Comment le faire ? Exemple : copier le texte ci-dessous dans un éditeur de texte (bloc-notes de Ms Office),

```
<livre>  
  <titre>Œuvre poétique </titre>  
  <auteur>Senghor, Léopold Sedar (1906-2001)</auteur>  
  <edition>Paris : Ed. du Seuil, 1990</edition>  
  <description>429 p. ; 22 cm</description>  
  <ISBN>2-02-012106-9</ISBN>  
  <prix>38,11 EUR</prix>  
</livre>
```

Ensuite l'enregistrer avec l'extension **.xml** (choisir votre propre nom de fichier) puis l'ouvrir avec Internet explorer (qui possède un parseur).

Document XML bien formé (suite)

Résultat : le fichier doit apparaître sous forme d'arborescence
comme ceci :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
_ <livre>
  <titre>Oeuvre poétique</titre>
  <auteur>Senghor, Léopold Sedar (1906-2001)</auteur>
  <edition>Paris : Ed. du Seuil, 1990</edition>
  <description>429 p. ; 22 cm</description>
  <ISBN>2-02-012106-9</ISBN>
  <prix>38,11 EUR</prix>
</livre>
```

NB : ISO-8859-1 est le jeu de caractères Latin 1 pour les langues d'Europe occidentale et d'Amérique latine



■ Question

comment s'assurer que les éléments sont utilisés de la même manière dans plusieurs documents XML du même type ?

■ Réponse

on associe aux documents XML une **DTD**
ou un **Schéma XML**, c'est-à-dire une
« grammaire commune »

Les DTD

DTD = document type description

- Permet de décrire l'arborescence-type des documents
- Permet de décrire le type de données pour chaque élément
- Le document XML est associé à la DTD
 - soit la DTD est interne au doc XML
 - soit la DTD est externe au doc XML

NB : une DTD n'est pas écrite en XML

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE livre SYSTEM "structure.dtd">
```

```
<livre>
  <titre>Œuvre poétique </titre>
  <auteur>Senghor, Léopold Sedar (1906-2001)</auteur>
  <edition>Paris : Ed. du Seuil, 1990</edition>
  <description>429 p. ; 22 cm</description>
  <ISBN>2-02-012106-9</ISBN>
  <prix>38,11 EUR</prix>
</livre>
```

poesie.xml

```
<!ELEMENT livre (titre+,auteur*,?,edition,description?, ISBN?,prix?)
  <!ELEMENT titre (#PCDATA)>
  <!ELEMENT auteur (#PCDATA)>
  <!ELEMENT edition (editeur,date)>
  <!ELEMENT editeur (#PCDATA)>
  <!ELEMENT date (#PCDATA)>
  <!ELEMENT description (#PCDATA)>
  <!ELEMENT ISBN (#PCDATA)>
  <!ELEMENT prix (#PCDATA)>
```

structure.dtd

```

<!ELEMENT livre (titre+,auteur*,?,edition,description?,ISBN?,prix?)
  <!ELEMENT titre (#PCDATA)>
  <!ELEMENT auteur (#PCDATA)>
  <!ELEMENT edition (editeur|date)>
  <!ELEMENT editeur (#PCDATA)>
  <!ELEMENT date (#PCDATA)>
  <!ELEMENT description (#PCDATA)>
  <!ELEMENT ISBN (#PCDATA)>
  <!ELEMENT prix (#PCDATA)>

```

DTD

Définition des éléments

,	séparateur d'une séquence d'éléments, ordre à respecter	
	séparateur d'une séquence d'éléments, pas d'ordre à respecter	
?	l'élément peut figurer 0 ou 1 fois	facultatif / non répétable
+	l'élément peut figurer 0 ou n fois	facultatif / répétable
*	l'élément peut figurer 1 ou n fois	obligatoire / répétable

PCDATA l'élément contient une chaîne de caractères

ANY l'élément peut contenir tout type de données

EMPTY l'élément est vide

Les schémas XML

- Ils ont la même fonction que la DTD : définir la structure-type d'un ensemble de documents XML
- Ils sont maintenus par le W3C
- Ils sont les « successeurs » des DTD
- avantages par rapport à la DTD :
 - sont écrits en XML
 - sont extensibles (peuvent intégrer d'autres schémas)
 - permettent des contraintes plus fines sur la forme des données

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<xs:element name="livre" type="livreType"/>
```

schema.xsd

```
<xs:complexType name="livreType">
```

```
<xs:sequence>
```

```
<xs:element name="titre" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
```

```
<xs:element name="auteur" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
```

```
<xs:element name="edition" type="editionType" minOccurs="0" maxOccurs="unbounded"/>
```

```
<xs:element name="description" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
```

```
<xs:element name="ISBN" type="ISBNType" minOccurs="0" maxOccurs="1"/>
```

```
<xs:element name="prix" type="xs:string" minOccurs="0" maxOccurs="1"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
<xs:complexType name="editionType">
```

```
<xs:sequence><xs:element name="editeur" type="xs:string"/><xs:element name="date" type="xs:string"/>
```

```
</xs:sequence></xs:complexType>
```

```
<xs:simpleType name="ISBNType"><xs:restriction base="xs:string"><xs:pattern value="\d-\d{5}-\d{3}-\d" />
```

```
</xs:restriction></xs:simpleType>
```

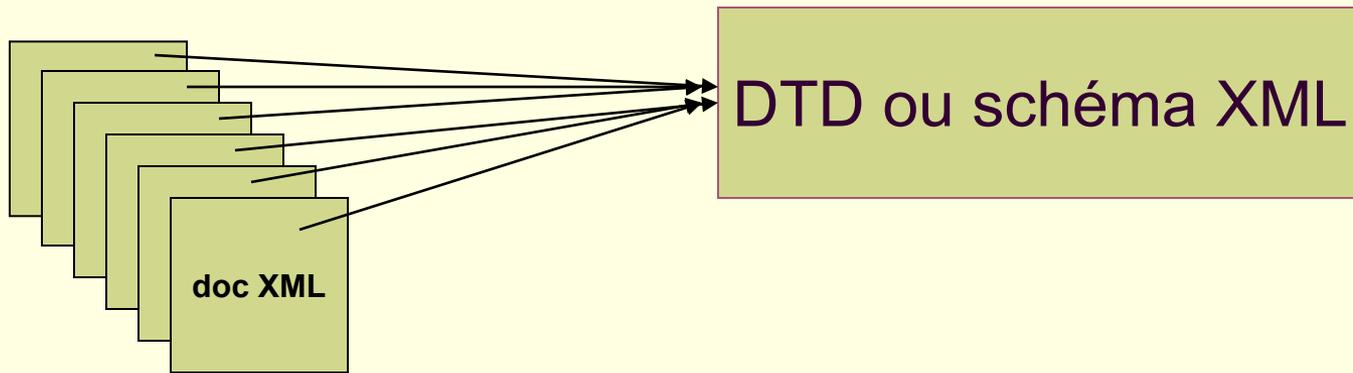
```
</xs:schema>
```

Les schémas XML

avec un schéma XML, on peut exercer un contrôle fin des valeurs

```
<xs:simpleType name="ISBNType"><xs:restriction  
  base="xs:string"><xs:pattern value="\d-\d{5}-\d{3}-\d" />  
</xs:restriction></xs:simpleType>
```

=>la valeur de l'élément **<ISBN>** doit être conforme au modèle défini dans le schéma : d-ddddd-ddd-d où chaque d est un chiffre



Une DTD ou un schéma XML peut s'appliquer à plusieurs documents XML

Un document XML conforme à une DTD ou un schéma est dit « **valide** »



■ Question

Comment définir la présentation graphique des données contenues dans un ou plusieurs document(s) XML ?

■ Réponse

En utilisant une feuille de style (*stylesheet*)

Les feuilles de style

Il existe deux langages pour en créer :

- **CSS**

- pour définir la présentation visuelle des informations contenues dans les balises XML

- **XSL-T**

- pour définir la présentation visuelle des informations contenues dans les balises XML
- pour transformer un arbre XML en un autre arbre XML

CSS

- **CSS** = cascading style sheets
 - feuille de style en cascade
- Maintenance assurée par le W3C
- La présentation graphique de chaque élément XML est définie de façon absolue, quelque soit sa position dans l'arbre
- CSS n'est pas écrit en XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet href="style1.css" type="text/css"?>
<livre>
<titre>Œuvre poétique </titre>
<auteur>Senghor, Léopold Sedar (1906-2001)</auteur>
<edition>Paris : Ed. du Seuil, 1990</edition>
<description>429 p. ; 22 cm</description>
<ISBN>2-02-012106-9</ISBN>
<prix>38,11 EUR</prix>
</livre>
```

poesie.xml

```
body { margin:15px 15px 15px 15px; }
```

```
titre { display:block; font-family:verdana, sans-serif; font-size:100%; color:white; font-weight:bold;
background-color:black; text-align:center; }
```

```
auteur { display:block; font-family:verdana, sans-serif; font-size:80%; color:black; background-
color:transparent; text-align:left;}
```

```
editeur { display:block; font-family:verdana, sans-serif; font-size:80%; color:black; background-
color:transparent; text-align:left; }
```

```
description { display:block; font-family:verdana, sans-serif; font-size:80%; color:black;
background-color:transparent; text-align:left; }
```

etc.

CSS



Œuvre poétique

Senghor, Léopold Sedar
Œuvre poétique
Paris : Ed. du Seuil
429 p. ; 22 cm
2-02-012106-9
38,11 EUR

Le résultat dans Internet Explorer

XSL-T

- **XSL-T** = eXtensible Stylesheet Language - Transformations
- Maintenance assurée par le W3C
- La présentation graphique des éléments s'effectue selon un arbre à définir (qui peut être différent de celui du document XML)
- XSL-T est écrit en XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <?xml-stylesheet href="style1.xsl" type="text/xsl"?>
  <livre>
    <titre>Œuvre poétique </titre>
    <auteur>Senghor, Léopold Sedar (1906-2001)</auteur>
    <edition>Paris : Ed. du Seuil, 1990</edition>
    <description>429 p. ; 22 cm</description>
    <ISBN>2-02-012106-9</ISBN>
    <prix>38,11 EUR</prix>
  </livre>
```

poesie.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<HTML><BODY><xsl:apply-templates /></BODY></HTML>
</xsl:template>

<xsl:template match="livre">
<table>
<tr><td bgcolor="black"><b><font size="3" face="arial" color="white">Titre : <xsl:value-of select="titre" />
</font></b></td></tr>
<tr><td><font size="2" face="arial" color="black">Auteur : <xsl:value-of select="auteur" /></font></td></tr>
<tr><td><font size="2" face="arial" color="black">Editeur : <xsl:value-of select="editeur" /></font></td>
</tr>
etc.
```

style1.xsl

XSL-T



Titre : Œuvre poétique

Auteur : Senghor, Léopold Sedar

Titre : Œuvre poétique

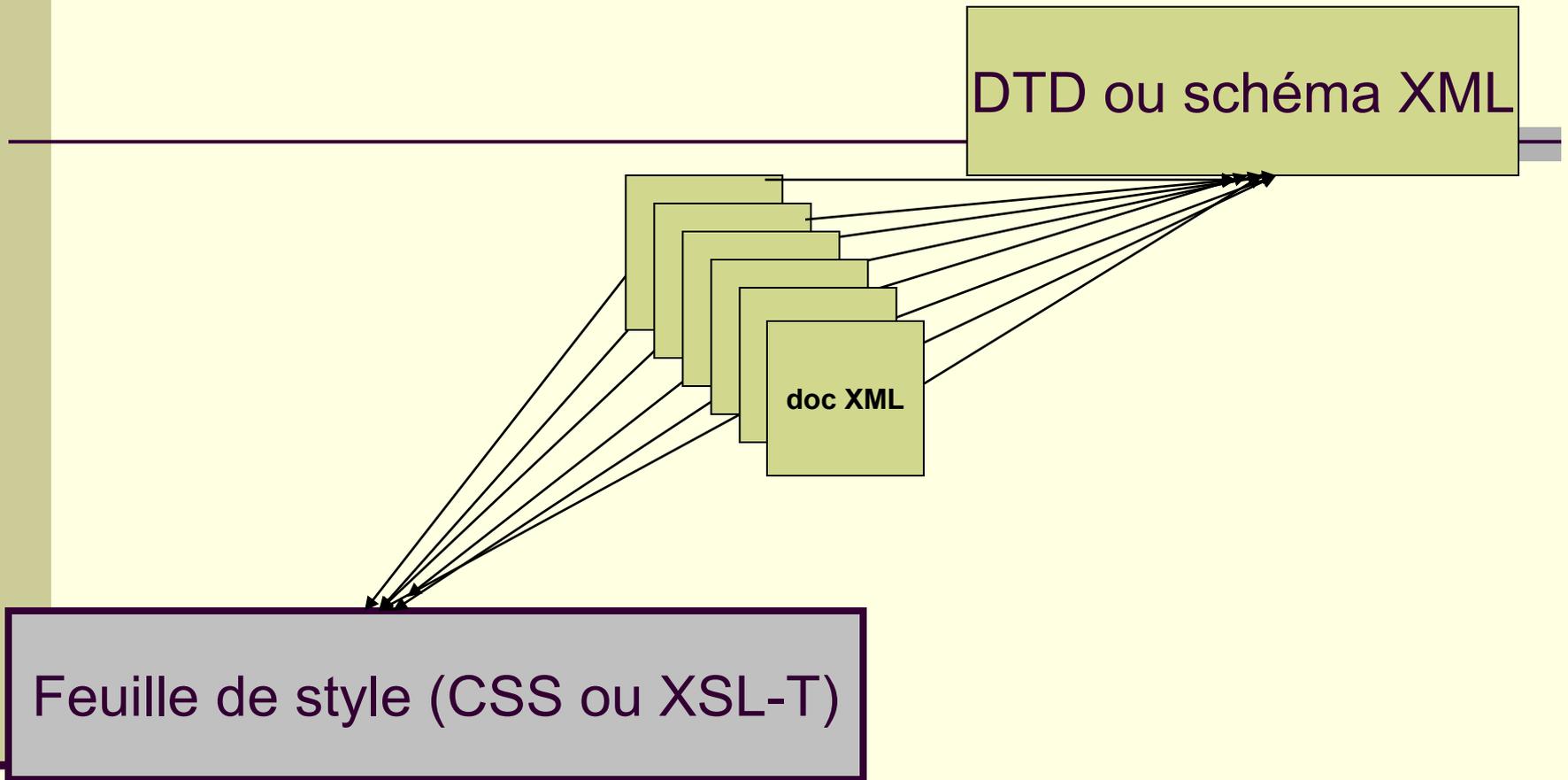
Editeur : Paris : Ed. du Seuil

Description : 429 p. ; 22 cm

ISBN : 2-02-012106-9

PRIX : 38,11 EUR

Le résultat dans Internet Explorer



Une feuille de style peut s'appliquer à plusieurs documents XML



■ Questions

- comment afficher des caractères « interdits » ou spéciaux ?
- comment utiliser des abréviations pour des éléments récurrents ?

■ Réponse

- en recourant aux *entités*

Les entités

Il en existe plusieurs types:

- Les entités prédéfinies
 - les entités XML
 - les entités caractères
- Les entités que l'on définit soi-même
 - les entités générales
 - les entités paramètres

(les entités générales et paramètres peuvent être internes ou externes)

Les entités prédéfinies

Les entités XML

< pour <

> pour >

" pour "

&mp; pour &

' pour '

Les entités caractères

î pour î

...

Les entités générales

- définies dans l'en-tête d'un doc XML ou dans une DTD externe
- définition de l'entité <!ENTITY nom_entité "chaîne_caractères">
appel de l'entité &nom_entité;

- exemple

```
<!ENTITY droits "Toute représentation ou reproduction, par quelque procédé que ce soit, notamment par
téléchargement ou sortie imprimante, constituera une contrefaçon sanctionnée par les articles L 335-2
et suivants du code de la propriété intellectuelle." >
```

...

```
<texte>&droits;</texte
```

...

à l'affichage de l'élément <texte> on aura :

Toute représentation ou reproduction, par quelque procédé que ce soit, notamment par téléchargement ou sortie imprimante, constituera une contrefaçon sanctionnée par les articles L 335-2 et suivants du code de la propriété intellectuelle.

Question : comment mettre cette phrase
entre " " à l'affichage ?

Les entités paramètres

- interprétées et remplacées dans la DTD elle-même

définition de l'entité	<code><!ENTITY %nom_entité "chaîne_caractères"></code>
appel de l'entité	<code>%nom_entité;</code>

- exemple

```
<ENTITY %codes-lang "EN|GE|JP|ES|FR" >
```

...

```
<!ATTLIST titreoriginal lang (%codes-lang;) >
```



■ Question

- comment éviter de confondre des éléments identiques (par ex. *titre*) mais utilisés dans des contextes différents ?

■ Réponse

- en utilisant les « espaces de noms » (*namespaces*)

A quel titre ?

Exemple de conflit

<livre>

<titre>Liberté III : Négritude et civilisation de l'universel </titre>

</livre>

...

<personne><nom>Senghor</nom><prenom>Léopold</prenom>

<titre>Membre de l'Académie française</titre> </personne>

...

=> problème : je veux utiliser les deux *titres* dans un même document XML. Comment l'application va-t-elle traiter la balise <titre> ?

Solution : définir deux contextes, deux espaces de noms

Les espaces de noms

- on crée un préfixe pour chaque contexte
 - par ex., « **li** » pour livre et « **ps** » pour personne
- on rattache ce préfixe à un URI
 - soit un URN
 - urn:ugb-bu:xml:livre
 - urn:ugb-bu:xml:personne
 - soit un URL
 - http://www.ugb-bu.sn/livre
 - http://www.ugb-bu.sn/personne

NB : l'URL sert d'identifiant unique, il ne pointe vers aucune ressource

Les espaces de noms

avec URN

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<livre xmlns:li="urn:ugb-bu:xml:livre" xmlns:ps="urn:ugb-bu:xml:personne">
  <li:titre>Liberté III : Négritude et civilisation de l'universel</li:titre>
  <auteur>
    <personne>
      <nom>Senghor </nom>
      <prenom>Léopold </prenom>
      <ps: titre>Membre de l'Académie française</ps: titre>
    </personne>
  </auteur>
</livre>
```

avec URL

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<livre xmlns:li="http://www.ugb-bu.sn/livre" xmlns:ps="http://ugb-bu.sn/personne">
  <li:titre>Liberté III : Négritude et civilisation de l'universel</li:titre>
  <auteur>
    <personne>
      <nom>Senghor </nom>
      <prenom>Léopold </prenom>
      <ps: titre>Membre de l'Académie française</ps: titre>
    </personne>
  </auteur>
</livre>
```

Espaces de noms

La galaxie XML

- **XPath** pour accéder à un élément dans un arbre XML
- **XLink** pour créer des liens entre fichiers XML
- **XPointer** pour créer des liens entre portions de fichiers XML
- **XML Schema** pour définir un arbre-type XML
- **XSL-T** pour afficher et transformer des éléments XML
- **XSL-FO** pour formater l'affichage et/ou l'impression d'un doc XML
- **XQuery** pour interroger sur des données XML
- **XFrames** pour afficher plusieurs pages dans un navigateur
- **XForms** pour créer des formulaires en ligne

Pour en savoir plus...

- <http://www.w3.org/> : Site du consortium du Web (W3C)
- <http://xmlfr.org/> (en français)
- <http://www.commentcamarche.net/xml/xmlintro.php3>
- Etc.

Ce document est publié sous licence creative commons.
Voir les [**conditions d'utilisation**](#)