

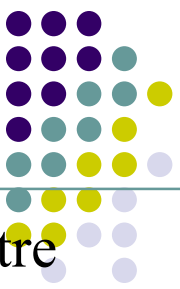


Rappels des notions de base de XML

Manipulation de données XML

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS





Séquence 1 : Rappels des notions de base de XML

Objectifs spécifiques : A la suite de ce chapitre, l'apprenant doit être capable de se rappeler de:

1. La définition du langage XML,
2. La description des différentes notions liées à XML,
3. La création d'un document XML,
4. Les technologies liées à XML

Plan :

- Présentation
- Structure d'un document XML
- Les règles de syntaxe
- Les éléments
- Les attributs
- Les espaces de nom (namespaces)
- Les technologies liées à XML

Présentation



- **Qu'est-ce que XML?**

- XML (eXtensible Markup Language) est un langage de balisage comme HTML et est conçu pour le stockage et le transport de données, et non pour l'affichage.
- Ses balises ne sont pas prédéfinies mais définies par l'utilisateur
- XML est une recommandation du W3C depuis le 10 Février 1998
- Son apprentissage requiert la connaissance de HTML, CSS et JavaScript

- **Que fait XML ?**

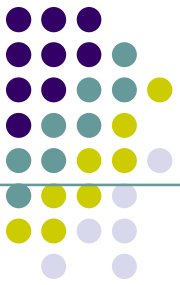
- En fait, XML décrit simplement de l'information en utilisant des balises
- Il faudra un logiciel pour le traitement (envoyer, recevoir ou afficher les données XML).
- Cependant XML permet d'apporter des simplifications significatives dans d'autres domaines.



Présentation

Exemple de document XML

```
<?xml version="1.0" encoding="UTF-8"?>
<notebook>
  <note>
    <destinataire>Caare</destinataire>
    <expediteur>Ngeer</expediteur>
    <titre>Rappel</titre>
    <message>N'oublies pas le cours de Xml!</message>
  </note>
  <note>
    <destinaire>saaliwu Géy</destinaire>
    <expediteur>Ngeer</expediteur>
    <titre>Information</titre>
    <message>Rencontre à la salle B17 à 10h30</message>
  </note>
</notebook>
```



Présentation

- **Apports de XML** : XML permet de simplifier :

Le partage des données

- En effet, de nombreux systèmes informatiques contiennent des données dans des formats incompatibles.
- L'échange de données entre les systèmes incompatibles (ou la mise à niveau de ces systèmes) constitue une tâche fastidieuse pour les développeurs web
- car de grandes quantités de données doivent être converties, et les données incompatibles sont souvent perdues.

Le transport des données

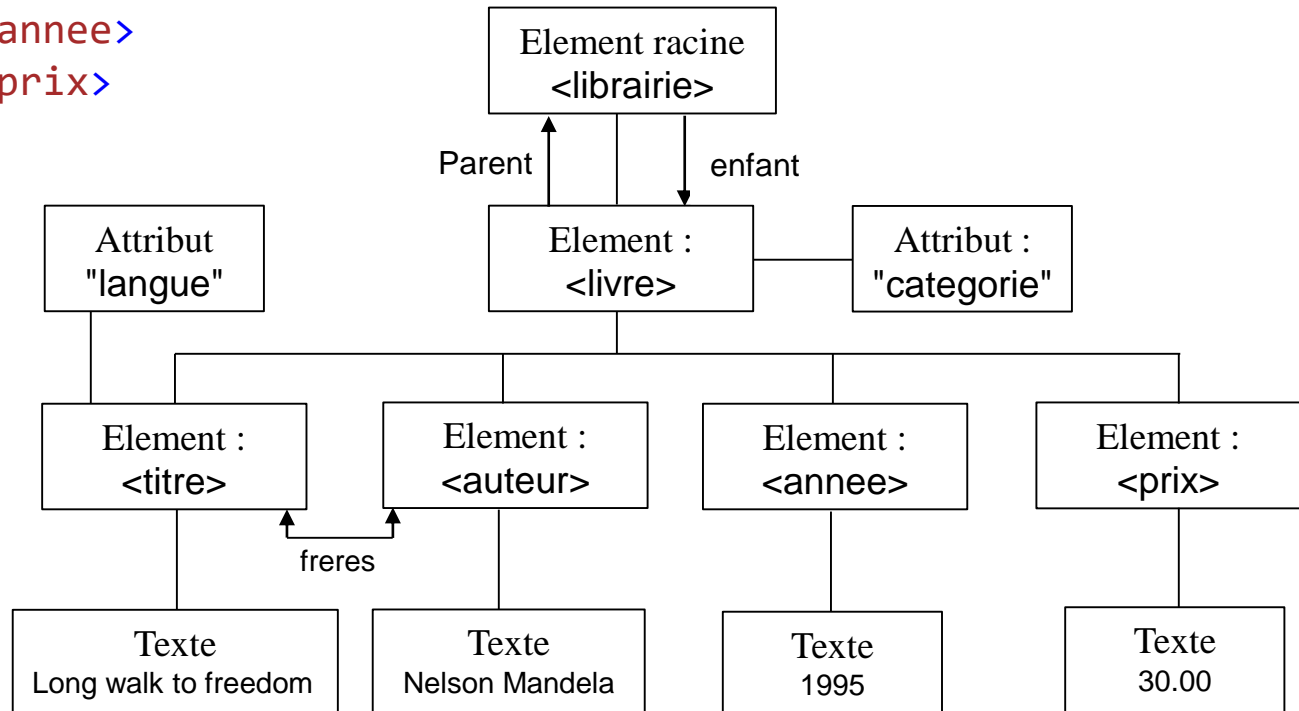
- En effet, XML stocke les données au format texte brut. Ce qui fournit, aux logiciels et matériels, un moyen indépendant de stockage, de transport et de partage des données.

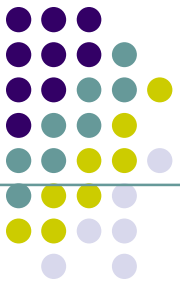


Structure d'un document XML

Les documents XML forment une structure arborescente qui commence à "la racine" et se ramifie en des «feuilles».

```
<?xml version="1.0" encoding="UTF-8"?>
<librairie>
  <livre categorie="Histoire">
    <titre langue="en">Long walk to freedom</titre>
    <auteur>Nelson Mandela</auteur>
    <annee>1995</annee>
    <prix>30.00</prix>
  </livre>
</librairie>
```





Structure d'un document XML

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE exemple SYSTEM "exemple.dtd">
<librairie>
  <livre categorie="Réseau">
    <titre lang="fr">Introduction au réseau mobile</ titre >
    <auteur>Alassane Diop</auteur>
    <annee>2011</annee>
    <prix>30.00</prix>
  </livre >
  <!--Un deuxieme livre-->
  <livre categorie="Compilation">
    <titre lang="en">Théorie des langages et automates</ titre >
    <auteur>E. M. Nguer</auteur>
    <annee>2012</annee>
    <prix>29.99</prix>
    <image src="/images/la.jpg" />
  </livre >
</librairie>
```

← Elément racine

← Elément

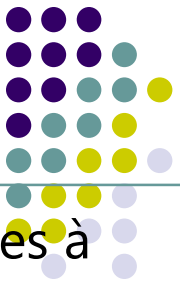
← Commentaire

← Attribut

← Elément vide

Entête

Contenu



Règles de syntaxe

Les règles de syntaxe de XML sont très simple et logique. Elles sont faciles à apprendre et à utiliser.

- **Un documents XML doit avoir un élément racine**
Cet élément **racine** est le **parent** de tous les autres éléments:

```
<racine>
  <enfant>
    <sousenfant>...</sousenfant>
  </enfant>
</racine>
```

Dans cet exemple, **<note>** est l'élément racine:

```
<?xml version="1.0" encoding="
UTF-8"?>
<note>
  <a>Caare</a>
  <de>Ngeer</de>
  <entete>Reminder</entete>
  <corps>Don't forget the XML
course!</corps>
</note>
```

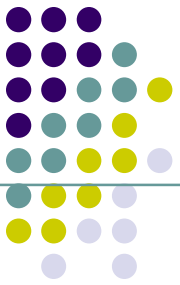



Règles de syntaxe

Le prologue XML

La ligne `<?xml version="1.0" encoding="UTF-8"?>` est appelée le **prologue** du document XML:

- Il est optionnel. S'il existe, il doit être en début du document.
- Un document XML peut contenir des caractères internationales, comme des caractères wolof ñó, norvégiens øæå ou français êèé.
 - Pour éviter des erreurs, vous devez spécifier l'encodage utilisé, ou sauvegarder vos fichiers XML en **UTF-8**.
 - **UTF-8** est l'encodage par défaut des caractères pour les documents XML, HTML5, CSS, JavaScript, PHP et SQL.



Règles de syntaxe

Balise de fermeture XML

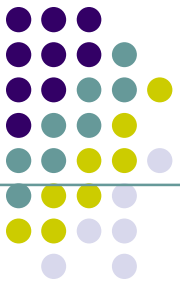
- En HTML, certains éléments peuvent bien fonctionner, même avec une balise de fermeture manquante:

```
<p>Ceci est un paragraphe.  
<br>
```

- En XML, il est illégal d'omettre la balise de fermeture. Tous les éléments doivent avoir une balise de fermeture:

```
<p>Ceci est un paragraphe.</p>  
<br />
```

- Le prologue XML n'a pas de balise de fermeture. Ce n'est pas une erreur, il fait pas partie du document XML.



Règles de syntaxe

Sensibilité à la casse des balises XML

- Les balises XML sont sensibles à la casse. La balise <Lettre> est différente de la balise <lettre>.
- Les balises d'ouverture et de fermeture doivent être écrites avec la même casse:

<Message>Ceci est incorrect </message>

<message>Ceci est correct </message>

"Les balises d'ouverture et de fermeture" sont souvent appelées "Balises de début et de fin".

Utilisez ce que vous préférez. C'est exactement la même chose.



Règles de syntaxe

L'imbrication des éléments en XML

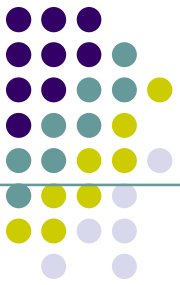
- En HTML, on peut avoir des éléments mal imbriqués :

`<i>`Ce texte est en gras et en italique`</i>`

- En XML, tous les éléments doivent être correctement imbriqués les uns dans les autres :

`<i>`Ce texte est en gras et en italique`</i>`

Dans l'exemple ci-dessus, "correctement imbriqué" signifie simplement que, puisque l'élément `<i>` est ouvert à l'intérieur de l'élément ``, il doit être fermé à l'intérieur de l'élément ``.



Règles de syntaxe

Citation des valeurs d'attribut

- Les éléments XML peuvent avoir des attributs sous forme de paires nom/valeur comme en HTML.
- En XML, les valeurs d'attributs doivent toujours être cités.

Incorrect

```
<note date=12/11/2007>  
  <a>Jaxabi</a>  
  <de>Jaañ</de>  
</note>
```

L'erreur dans ce document est que la valeur de l'attribut date dans l'élément note n'est pas citée.

Correct

```
<note date="12/11/2007">  
  <a>Jaxabi</a>  
  <de>Jaañ</de>  
</note>
```



Règles de syntaxe

Références d'entité

- Certains caractères ont une signification particulière en XML. Si vous placez un caractère comme "<" à l'intérieur d'un élément XML, il va générer une erreur parce que l'analyseur l'interprète comme le début d'un nouvel élément.

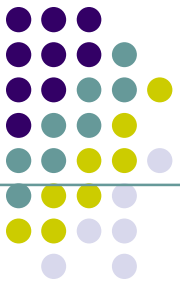
Exemple : `<message>salaire < 1000</message>`

- Pour éviter cette erreur, on remplace le caractère "<" par une **référence d'entité** : `<message>salaire < 1000</message>`

Il y a 5 références d'entité prédéfinies en XML:

Seuls < et & sont strictement illégale en XML, mais il est une bonne habitude de remplacer > par >.

<	<	less than	Inferieur
>	>	greater than	superieur
&	&	ampersand	esperluette
'	'	apostrophe	apostrophe
"	"	quotation mark	Guillemet



Règles de syntaxe

Commentaires en XML

- La syntaxe des commentaires en XML est similaire à celle en HTML.

`<!--Ceci est un commentaire-->`

Deux tirets dans un commentaire n'est pas autorisé.

Non autorisé: `<!-- Ceci est un -- commentaire -->`

Étrange, mais admis: `<!--Ceci est un - - commentaire -->`

Espace blanc en XML

XML ne tronque pas plusieurs espaces blancs (HTML tronque plusieurs espaces blancs en un espace blanc simple):

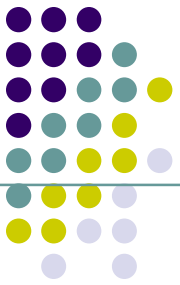
XML:	Siyaar Bàmba
HTML:	Siyaar Bàmba



Règles de syntaxe

Document XML bien formé

Un document XML conforme aux règles de syntaxe énoncées ci-dessus est dit "bien formé".



Les éléments

Un document XML contient des éléments XML.

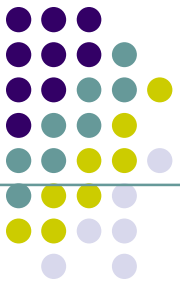
Qu'est-ce qu'un élément XML?

Un élément XML est tout ce qui est compris entre une balise de début et sa balise de fin (y compris la balise).

```
<prix>29.99</prix>
```

Un élément XML peut contenir :

- du texte
- des attributs
- d'autres éléments
- ou un mélange de ce qui précède



Les éléments

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>  
<!DOCTYPE exemple SYSTEM "exemple.dtd">
```

Entête

```
<librairie>
```

← **Elément racine**

```
<livre categorie="Réseau">
```

← **Elément**

```
<titre lang="fr">Introduction au réseau mobile</ titre >
```

```
<auteur>Alassane Diop</auteur>
```

```
<annee>2011</annee>
```

```
<prix>30.00</prix>
```

```
</livre >
```

```
<!--Un deuxieme livre-->
```

← **Commentaire**

```
<livre categorie="Compilation">
```

```
<titre lang="en">Théorie des langages et automates</ titre >
```

```
<auteur>E. M. Nguer</auteur>
```

← **Attribut**

```
<annee>2012</annee>
```

```
<prix>29.99</prix>
```

```
<image src="/images/la.jpg" />
```

← **Elément vide**

```
</livre >
```

```
</librairie>
```

Contenu



Les éléments

Dans l'exemple ci-dessus :

- `<titre>`, `<auteur>`, `<annee>` et `<prix>` ont du **contenu texte**, car ils contiennent du texte (comme 29.99).
- `<librairie>` et `<livre>` ont des contenus éléments, car ils contiennent des éléments.
- `<livre>` a un attribut (`categorie = "Réseau"`)

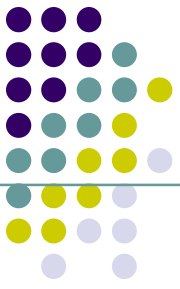
Éléments XML vides

Un élément sans contenu est dit vide. En XML, on peut indiquer un élément vide comme suit :

`<element></element>`

On peut également utiliser une balise dite auto-fermée: `<element />`

Les deux formes produisent des résultats identiques dans un logiciel XML (Lecteurs, Parseurs, Navigateurs). Un éléments vide peut avoir des attributs.



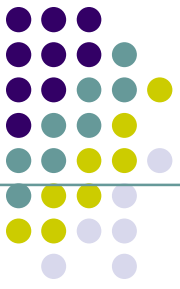
Les éléments

Règles de nommage en XML

Un élément XML doit suivre les règles de nommage suivantes:

- Il est sensible à la casse
- Il doit commencer par une lettre ou un trait de soulignement
- Il ne peut pas commencer par les lettres xml (ou XML, ou Xml, etc)
- Il peut contenir des lettres, des chiffres, des tirets de soulignement, et des signes de ponctuation.
- Il ne peut pas contenir d'espaces

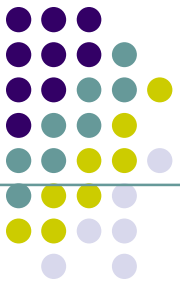
Tout nom peut être utilisé, pas de mots sont réservés (**sauf xml**).



Les éléments

Meilleures pratiques en nommage

- Créer des noms descriptifs comme ceci: `<personne>`, `<prenoms>`
- Créer des noms courts et simples comme ceci: `<titre_livre>` et pas comme: `<le_titre_du_livre>`.
- Eviter "-". Si vous nommez quelque chose par "titre-livre", un logiciel peut penser que vous voulez soustraire "livre" de "titre".
- Eviter ".". Si vous nommez quelque chose par "titre.livre", un logiciel peut penser que "livre" est une propriété de l'objet "titre".
- Eviter ":". Les : sont réservés aux espaces de noms (plus tard).
- Les lettres non anglaises comme éòá sont parfaitement légales en XML, mais attention aux problèmes si votre logiciel ne les prend pas en charge.



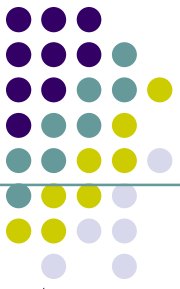
Les éléments

Styles de nommage

- Il n'y a pas de styles de nommage définis pour les éléments XML. Mais voici les couramment utilisés:

Style	Example	Description
Lower case	<firstname>	Toutes les lettres minuscules
Upper case	<FIRSTNAME>	Toutes les lettres majuscules
Underscore	<first_name>	Le trait de soulignement sépare les mots
Pascal case	<FirstName>	Premiere letter de chaque mot en majuscule
Camel case	<firstName>	Premiere letter de chaque mot en majuscule excepté le premier le mot.

- Si vous choisissez un style de nommage, il est bon d'être cohérent!
Les documents XML ont souvent une base de données correspondante. Il est bien d'utiliser les mêmes règles de nommage dans la base de données.



Les attributs

Les éléments XML peuvent avoir des attributs, tout comme HTML.
Les attributs sont conçus pour contenir des données relatives à un élément spécifique.

- **Citation des attributs XML :**

Elle se fait par des apostrophes (') ou des guillemets (").

Exemple :

`<personne genre="femme">` ou comme suit:

`<personne genre='homme'>`

Si la valeur de l'attribut elle-même contient des guillemets, vous pouvez utiliser des apostrophes, comme dans cet exemple:

`<roi nom='Lat Joor "Ngóone Laatir" Jóob'>`

Ou des entités :

`<roi nom='Lat Joor "Ngóone Latiir" Jóob'>`



Les espaces de noms

Les espaces de noms fournissent une méthode d'éviter les conflits de noms d'élément.

Conflits de nom

En XML, les noms des éléments sont définis par le développeur. Ce qui entraîne souvent des conflits quand on mélange des documents XML provenant de d'applications XML différentes.

Considerons les deux tables XML suivantes :

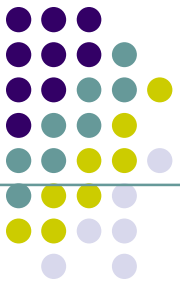
1- Informations sur un tableau HTML

```
<table>
  <tr>
    <td>Pommes</td>
    <td>Bananes</td>
  </tr>
</table>
```

2- Informations sur une table

```
<table>
  <name>table de café
    africain</name>
  <width>80</width>
  <length>120</length>
</table>
```

Ces deux fragments de code, mélangés ensemble, pourraient générer des conflits de noms. En effet, tous les deux contiennent un élément `<table>`, mais avec chacun un contenu et une signification différents. Un utilisateur ou une application XML ne pourra pas être en mesure de traiter ces différences.



Les espaces de noms

Résoudre les conflits de noms en utilisant un préfixe

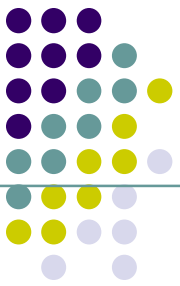
Les conflits de nom en XML peuvent facilement être évités en utilisant un préfixe de nom.

Le document suivant porte des données sur un tableau HTML, et des données sur une table de produits :

```
<h:table>
  <h:tr>
    <h:td>Pommes</h:td>
    <h:td>Bananes</h:td>
  </h:tr>
</h:table>
```

Dans cet exemple, il n'y aura pas de conflit parce que les deux éléments `<table>` ont des noms différents.

```
<f:table>
  <f:nom>Table de Café Africain</f:nom>
  <f:largeur>80</f:largeur>
  <f:longueur>120</f:longueur>
</f:table>
```



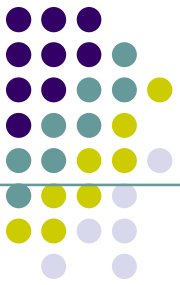
Les espaces de noms

L'attribut xmlns

Lors de l'utilisation des préfixes, un espace de noms pour le préfixe doit être défini. Il est défini par l'attribut xmlns dans la balise de début comme suit :

Syntax : `xmlns:prefix="URI"`.

```
<racine>
<h:table xmlns:h="http://www.ugb.sn/TR/html4/">
  <h:tr>
    <h:td>Pommes</h:td>
    <h:td>Bananes</h:td>
  </h:tr>
</h:table>
<f:table xmlns:f="http://www.marchesor.com/fourniture">
  <f:nom>Table de Café Africain </f:nom>
  <f:largeur>80</f:largeur>
  <f:longueur>120</f:longueur>
</f:table>
</racine>
```



Les espaces de noms

L'attribut xmlns

Dans l'exemple ci-dessus:

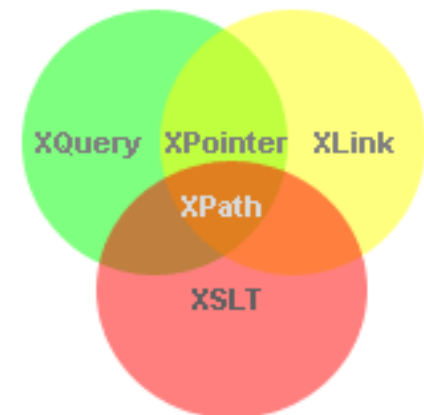
- L'attribut xmlns dans le premier élément <table> fait du préfixe h: un espace de noms qualifié.
- L'attribut xmlns dans le second élément <table> fait du préfixe f: un espace de noms qualifié.
- Quand un espace de noms est défini pour un élément, tous les éléments enfant avec le même préfixe sont associés avec le même espace de noms.

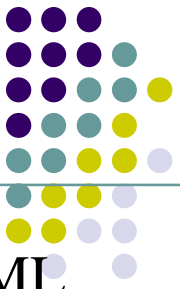


Les technologies liées à XML

Il existe plusieurs technologies utilisées pour le traitement d'un document XML. On peut en citer :

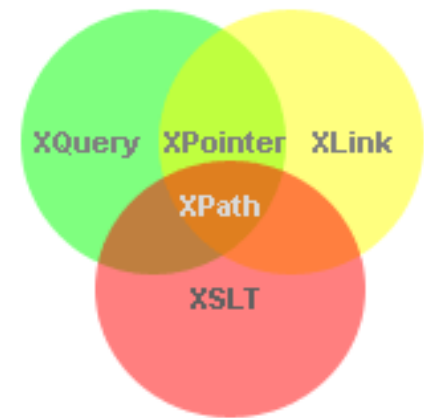
- Les langages **DTD** et **XML SCHEMA** pour la définition du type de document XML.
- **CSS** et **XSLT** pour le formatage d'un document XML
- **XPath** pour trouver des informations dans un document XML.
- **Xquery** qui est le langage de requête pour XML





Les technologies liées à XML

- **Xlink** qui est utilisé pour créer des hyperliens dans les documents XML
- **Xpointer** qui permet aux liens de pointer vers des parties spécifiques d'un document XML
- **JavaScript** pour l'affichage d'un document XML au niveau du client
- **PHP** pour l'affichage d'un document XML au niveau du serveur
- Toutes ces technologies seront étudiées dans la suite du cours





Fin de la séquence 1