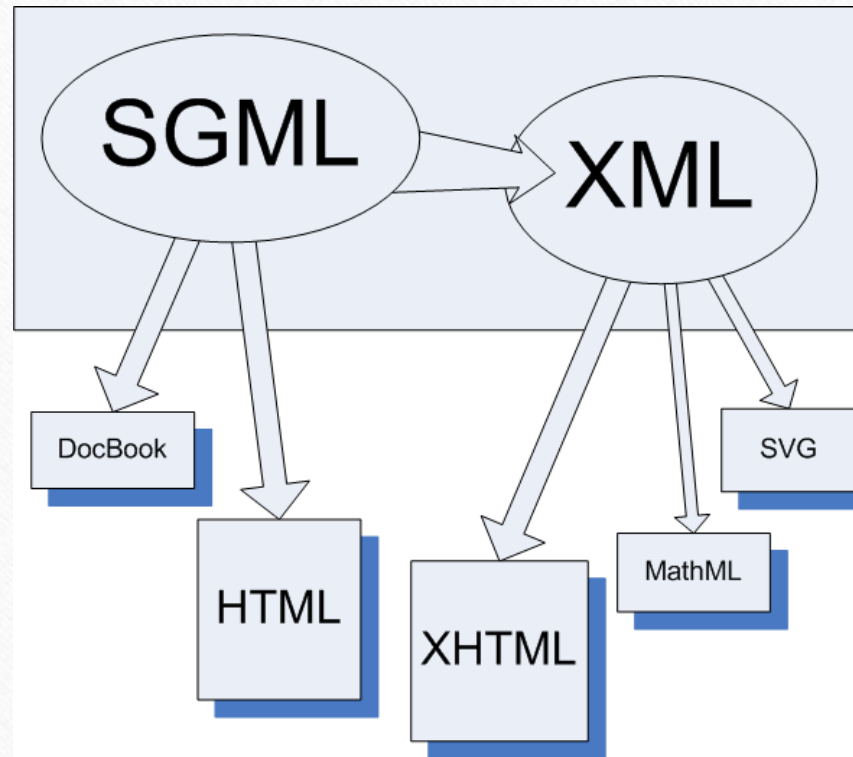


XML

Introduction & Syntaxe de XML

Par **Dr. Ibrahima DIOP**,
Enseignant – Chercheur à
L'Université Assane SECK de Ziguinchor

SGML-HTML-XML-XHTML

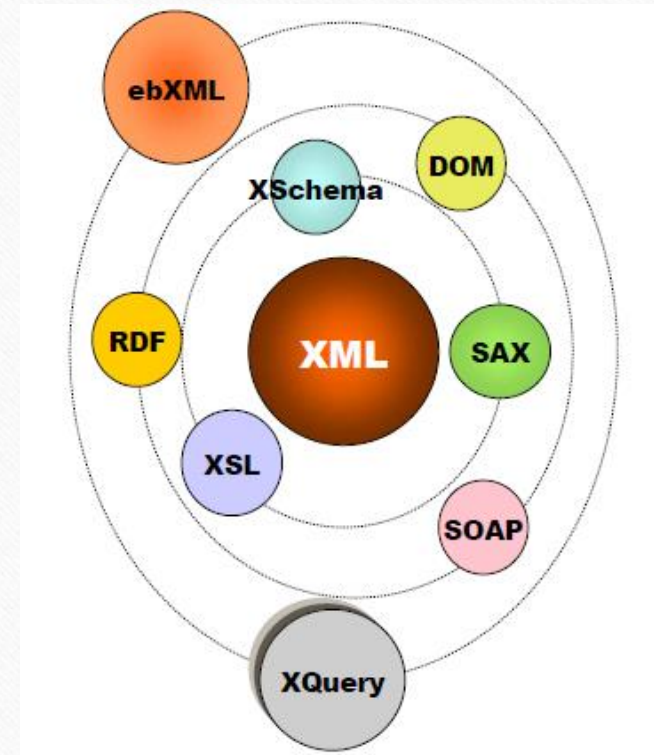


Les qualités de XML

- Les caractéristiques essentielles qui ont conduit aux succès de XML :
 - (1) **Séparation stricte entre contenu et présentation**
 - (2) **Structuration forte**
 - (3) **Extensibilité et flexibilité**
 - (4) **Modèles de documents**
 - (5) **Gestion des caractères spéciaux**
 - (6) **Simplicité**
 - (7) **Format libre**

La galaxie XML

- Xpath
 - pour la recherche d'éléments
- XSL
 - pour la transformation
 - et l'affichage
- Xquery
 - pour l'interrogation
- DOM
 - pour la programmation
- ...



Orthographe et la grammaire

- L'orthographe est constitué de règles pour la bonne écriture des mots.
- La grammaire régit l'agencement des mots dans une phrase.
- Pour qu'une phrase dans une langue soit correcte, il faut d'abord que les mots soient bien orthographiés et, ensuite, que la phrase soit bien construite.
- XML a également ces deux niveaux.

Syntaxe et structure de XML

- Pour qu'un document XML soit correct, il doit d'abord être **bien formé** et, ensuite, être **valide**.
- Un document **bien formé** doit respecter certaines règles **syntaxiques** propres à XML (orthographe d'XML).
- Un document **valide** doit respecter un modèle de document qui décrit de manière rigoureuse comment doit être **structurer** le document.

Parties d'un document XML

- Un document XML a les deux parties suivantes :
 - Prologue
 - Corps

<code><?xml ... ?></code>	<code>]</code>	Prologue
<code>...</code>	<code>]</code>	
<code><root-element></code>	<code>]</code>	
<code>...</code>	<code> </code>	Corps
<code></root-element></code>	<code>]</code>	

Prologue

- Le prologue peut contenir :
 - une déclaration XML (Entête XML)
 - une DTD ou un schéma

<code><?xml ... ?></code>]	Entête XML]
<code><!DOCTYPE root-element [</code>]		Prologue
<code>...</code>		DTD	
<code>]></code>]]

Entête XML

`<?xml version="1.0" encoding='iso-8859-1' standalone='yes'?>`

Version du langage : 1.0

Encoding :

- jeu de codage de caractères utilisé dans le document.
 - Le jeu de caractères standard pour la France est le ISO-8859-1
 - Par défaut, l'attribut encoding a la valeur UTF-8 (Unicode)

Standalone

- Si Standalone = 'yes', le processeur considère que toutes les déclarations nécessaires au traitement du document sont incluses dans le document courant
 - **le document est autonome et ne requiert aucune autre donnée externe**

Déclaration du type de document (DTD)

- Un DTD (*Document Type Definition*) est une déclaration optionnelle qui sert à attacher une grammaire à un document XML.
- Elle est introduite avant la première balise (élément racine) de votre document sous cette forme :
- **<!DOCTYPE élément racine SYSTEM "URI vers la DTD">**
- Exemple :
 - **<!DOCTYPE cours SYSTEM "cours.dtd">**

Corps du document XML

...] Commentaires et instructions de traitement]	
<root-element>] Balise ouvrante de l'élément racine		Corps
...] Éléments , commentaires et		du
...] instructions de traitement		document
</root-element>] Balise fermante de l'élément racine		
...] Commentaires et instructions de traitement]	

Les commentaires

- Ce sont les mêmes qu'en HTML.
- Exemple:
 - **<!-- Date de création : 30/09/07 -->**
- Point important : les caractères -- sont interdits comme commentaires.
- Un commentaire est placée en dors des éléments

Les éléments

- Les éléments gèrent la structuration des données d'un document XML
 - un peu à la manière des répertoires qui servent à l'organisation des fichiers.
- On peut les qualifier de métadonnées, au sens où ils ne font pas partie réellement des données mais servent à en désigner la nature.
- À la place du terme élément, on peut utiliser les termes **balise**, *tag* ou encore **nœud**.

Les éléments

<?xml version="1.0" encoding="ISO-8859-1"?>

<cours>

<intervenant> Ibrahima Diop </intervenant>

<separateur/>

<chapitre>

Formation XML

<para>Un paragraphe</para>

<para>Autre paragraphe</para>

</chapitre>

</cours>

élément racine contenant trois éléments fils : intervenant, separateur et chapitre

Les éléments

<?xml version="1.0" encoding="ISO-8859-1"?>

<cours>

<intervenant> Ibrahima Diop </intervenant>

<separateur/>

<chapitre>

Formation XML

<para>Un paragraphe</para>

<para>Autre paragraphe</para>

</chapitre>

</cours>

élément contenant du texte

Les éléments

<?xml version="1.0" encoding="ISO-8859-1"?>

<cours>

<intervenant> Ibrahima Diop </intervenant>

<separateur/>

<chapitre>

Formation XML

<para>Un paragraphe</para>

<para>Autre paragraphe</para>

</chapitre>

</cours>

élément sans contenu

Les éléments

<?xml version="1.0" encoding="ISO-8859-1"?>

<cours>

<intervenant> Ibrahima Diop </intervenant>

<separateur/>

<chapitre>

Formation XML

<para>Un paragraphe</para>

<para>Autre paragraphe</para>

</chapitre>

</cours>

élément contenant du texte et des éléments fils para

Les éléments

<?xml version="1.0" encoding="ISO-8859-1"?>

<cours>

<intervenant> Ibrahima Diop </intervenant>

<separateur/>

<chapitre>

Formation XML

<para>Un paragraphe</para>

<para>Autre paragraphe</para>

</chapitre>

</cours>

élément contenant du texte

Les attributs d'un élément

- Un attribut est un couple (clé, valeur) associé à la définition d'un élément.
- L'attribut est complémentaire de l'élément de par son rôle au sens où il ajoute une information à l'élément.
- Exemple :
 - `<auteur nom="DIOP" prenom="Mamadou">...</auteur>`
 - `<contact email='mdiop@gmail.com' />`

Choix entre éléments et attributs

- Cas avec attributs :

`<personne nom="FALL" prenom="Mamadou"/>`

- Cas sans attribut :

`<personne>`

`<nom>FALL</nom>`

`<prenom>Mamadou</prenom>`

`</personne>`

- Cependant, l'inverse n'est pas vrai car un attribut ne peut pas être répété dans un élément.

Choix entre éléments et attributs

- Cas sans attribut :

<personne>

<nom>FALL</nom>

<prenom>Mamadou</prenom>

<prenom>Lamine</prenom>

</personne>

- Cas avec attributs :

<personne nom="FALL" prénom="Mamadou"
prénom="Lamine"/>

Impossible car deux attributs de même nom

Choix entre éléments et attributs

règles simples d'utiliser un attribut ou un élément :

- Lorsqu'une valeur est de **taille modeste**, a **peu de chance d'évoluer** vers une structure plus complexe, et **n'est pas répétée**, **alors l'attribut peut tout à fait convenir**.
- Dans tous les autres cas, **l'élément reste incontournable**. Car l'élément est le plus ouvert et garantira le plus facilement une évolution.

Les nœuds textes

- Le texte qui est associé à l'attribut (sa valeur) ou à l'élément (son contenu) sont les données.
- Exemple :

```
<cours titre="XML">
```

```
...
```

```
<plan>
```

```
    Introduction
```

```
...
```

```
</plan>
```

```
</cours>
```


Les nœuds textes

- Les données constituent le cœur du document.
- Tout le reste ne sert qu'à séparer et classer ces données.
- Les données sont dites terminales dans l'arborescence XML.

Entités prédéfinies

- Comme certains caractères sont réservés à la syntaxe XML, il faut être vigilant lors de l'écriture des données.
- Exemple :
 <calcul>
 if (a<b et b>c) ...
 </calcul>

Entités prédéfinies

- Les entités prédéfinies ne servent qu'à lever une ambiguïté syntaxique pour le parseur.
- Ces entités peuvent être utilisées dans un élément (pour du texte) ou dans un attribut (pour sa valeur).

Entités prédéfinies

- Voici la liste des entités prédéfinies :
 - < équivalent de < (*less than*) ;
 - > équivalent de > (*greater than*) ;
 - & équivalent de & (*ampersand*) ;
 - " équivalent de " (*quote*) ;
 - ' équivalent de ' (*apostrophe*).
- L'exemple précédant peut être réécrit :

```
<calcul>  
  If (a&lt;b et b&gt;c)...  
</calcul>
```

***CDATA**(Character Data)*

- un bloc de texte dont les caractères seront pris tel quel par le parseur jusqu'à la séquence de fin `]]>`.

- Exemple :

`<![CDATA[`

`<element>`

C'est un document XML

`</element>`

`]]>`

Les entités du document

- Dans une DTD (*Document Type Definition*), on peut créer des entités et y faire référence dans des documents XML.

- Exemple:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE cours [
```

```
<!ENTITY auteur "Ibrahima DIOP">
```

```
<!ELEMENT cours (#PCDATA)>
```

```
]>
```

```
<cours>
```

```
  Cours réalisé par &auteur;
```

```
</cours>
```


Syntaxe de XML - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!-- Date de création : 14/10/2014 -->
```

```
<!DOCTYPE cours SYSTEM "cours.dtd">
```

```
<cours titre="XML">
```

```
  <intervenant nom="Ibrahima Diop">
```

```
  </intervenant>
```

```
  <plan>
```

```
    Introduction
```

```
    XML et la composition de documents
```

```
  </plan>
```

```
</cours>
```

Syntaxe de XML - Exemple

Entête XML avec la version 1.0 et l'encodage iso-8859-1 des caractères.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!-- Date de création : 14/10/2014 -->
```

```
<!DOCTYPE cours SYSTEM "cours.dtd">
```

```
<cours titre="XML">
```

```
  <intervenant nom="Ibrahima Diop">
```

```
  </intervenant>
```

```
  <plan>
```

```
    Introduction
```

```
    XML et la composition de documents
```

```
  </plan>
```

```
</cours>
```

Syntaxe de XML - Exemple

<?xml version="1.0" encoding="UTF-8" ?> Commentaire délimité par les chaînes de caractères <!-- et -->.

<!-- Date de création : 14/10/2014 -->

<!DOCTYPE cours SYSTEM "cours.dtd">

<cours titre="XML">

 <intervenant nom="Ibrahima Diop">

 </intervenant>

 <plan>

 Introduction

 XML et la composition de documents

 </plan>

</cours>

Syntaxe de XML - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!-- Date de création : 14/10/2014 -->
```

```
<!DOCTYPE cours SYSTEM "cours.dtd">
```

```
<cours titre="XML">
```

Balise ouvrante de l'élément racine cours

```
  <intervenant nom="Ibrahima Diop">
```

```
  </intervenant>
```

```
  <plan>
```

```
    Introduction
```

```
    XML et la composition de documents
```

```
  </plan>
```

```
</cours>
```

Syntaxe de XML - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!-- Date de création : 14/10/2014 -->
```

```
<!DOCTYPE cours SYSTEM "cours.dtd">
```

```
<cours titre="XML">
```

```
  <intervenant nom="Ibrahima Diop">
```

```
  </intervenant>
```

```
  <plan>
```

```
    Introduction
```

```
    XML et la composition de documents
```

```
  </plan>
```

```
</cours>
```

Balise ouvrante de l'élément intervenant avec un attribut nom

Syntaxe de XML - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!-- Date de création : 14/10/2014 -->
```

```
<!DOCTYPE cours SYSTEM "cours.dtd">
```

```
<cours titre="XML">
```

```
  <intervenant nom="Ibrahima Diop">
```

```
  </intervenant>
```

```
  <plan>
```

```
    Introduction
```

```
    XML et la composition de documents
```

```
  </plan>
```

```
</cours>
```

Balise fermante de l'élément racine cours

DTD - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!-- Date de création : 14/10/2014 -->
```

```
<!DOCTYPE cours SYSTEM "cours.dtd">
```

Déclaration du Document Type Definition (DTD)

```
<cours titre="XML">
```

```
  <intervenant nom="Ibrahima Diop">
```

```
  </intervenant>
```

```
  <plan>
```

```
    Introduction
```

```
    XML et la composition de documents
```

```
  </plan>
```

```
</cours>
```

DTD - Exemple

Le document cours.dtd contient :

- `<!DOCTYPE cours [`
- `<!ELEMENT cours (intervenant, plan)>`
- `<!ELEMENT intervenant EMPTY>`
- `<!ELEMENT plan (#PCDATA)>`
- `<!ATTLIST cours titre CDATA #REQUIRED>`
- `<!ATTLIST intervenant nom CDATA #REQUIRED>`
- `]>`

Quelques règles de syntaxe

- Le nom d'un élément ne peut pas commencer par un chiffre.
- Si le nom d'un élément est composé d'un seul caractère il doit être dans la plage **[a-zA-Z]** ou **_** ou **:**.
- Avec au moins 2 caractères, le nom d'un élément peut contenir **_**, **-**, **.** et **:** plus les caractères alphanumériques.

Quelques règles de syntaxe

- Tous les éléments ouverts doivent être fermés.
- Un élément parent est toujours fermé après la fermeture des éléments fils.
 - Voici un contre-exemple : `<a>`.

Quelques conventions de nommage

- **Employer des minuscules** pour les attributs et les éléments.
- **Éviter les accents** dans les noms d'attributs et d'éléments pour des raisons de compatibilité avec les outils du marché qui proviennent souvent d'un univers anglo-saxon.
- **Séparer les noms** composés de plusieurs mots par les caractères -, __, . ou **une majuscule**.
- **Essayer d'être homogène** dans votre document en gardant la même convention.

FIN.
