Présentation de XML

Cours DAX pour Master-2 option TI

Présentation de XML

Définition, Historique, intérêts, syntaxe XML….

Définition

XML (e**X**tended **M**arkup **L**anguage) un métalangage de balisage, conçu vers 1997 afin de faciliter l'échange de données via le Web.

- Langage orienté texte, formé de balises permettant d'organiser les données de manière structurée;
- Standard incontournable de l'informatique, utilisé aussi bien pour le stockage et le partage de documents, que pour la transmission de données entre applications;
- Simple, flexible et facile d'utilisation, format universel de texte brut, auto-descriptif, normalisé et ouvert;
- Adapté aux traitements automatiques, disponibilité d'outils génériques d'analyse et de transformation.

Historique

XML dérive du langage **SGML** (**S**tandard **G**eneralized **M**arkup **L**anguage), développé dans les années 80. Ce langage, servant à préciser la structure d'un document quelconque, a été conçu pour les documentations techniques de grande ampleur. Sa généralité la rendue difficile et complexe, ce qui a freiné son utilisation.

Le **HTML** (**H**yper**T**ext **M**arkup **L**anguage) une version allégée et adapté à l'écriture de documents pour Internet a été développée. Mais ce dernier reste limité malgré de nombreuses adaptations. C'est alors que fut créé le **XML**.

Le **XML** est un dérivé du **SGML**. Il se sert des principes de simplicité du **HTML** et de la souplesse **SGML**.

Intérêts

Le succès du XML est en grande partie dû à ses qualités qui sont essentiellement :

- Séparation stricte entre contenu et présentation
- Simplicité, universalité et extensibilité
- Format texte avec gestion des caractères spéciaux
- Structuration forte
- Modèles de documents (DTD et Schémas XML)
- Format libre

Langages apparentés : Un des atouts indéniables de XML est le nombre de technologies et de langages qui se sont développés autour de XML. Les principaux langages sont :

- XLink et XPointer (liens entre documents)
- XPath (langage de sélection)
- XQuery (langage de requête)
- Schémas XML (modèles de documents)
- XSLT (transformation de documents)

Présentation de XML

Syntaxe XML

Il y a, en français, l'orthographe et la grammaire. La première est constituée de règles pour la bonne écriture des mots. La seconde régit l'agencement des mots dans une phrase. Pour qu'une phrase en français 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. Pour qu'un document XML soir correct, il doit être :

- ***Bien formé*** : Un document bien formé doit respecter certaines règles syntaxiques propres à XML. Il s'agit, en quelque sorte, de l'orthographe d'XML.
- ***Valide*** : Un document valide doit respecter un *modèle de document* qui décrit de manière rigoureuse comment doit être organisé le document. Un modèle de documents peut être vu comme une grammaire pour des documents XML.

La différence essentielle avec le français est que la grammaire d'XML n'est pas figée. Pour chaque application, il est possible de choisir la grammaire la plus appropriée. Cette possibilité d'adapter la grammaire aux données confère une grande souplesse à XML.

La syntaxe de XML est relativement simple. Elle est constituée de quelques règles pour l'écriture d'un entête et des balises pour structurer les données. Ces règles sont très similaires à celles du langage HTML utilisé pour les pages WEB mais elles sont :

- ***Plus générales*** : car les noms des balises sont libres.
- ***Plus strictes*** : car elles imposent qu'à toute balise ouvrante corresponde une balise fermante.

NB : La norme XML n'impose pas l'utilisation d'un modèle de document qui décrit comment doit être organisé le document, mais elle impose par contre le respect exact des règles de base de la norme XML.

Contenu

Un document XML contient:

- Un **prologue :** sa présence est facultative, mais fortement conseillée. Contient toutes les informations autres que les données ou les éléments;
- Un **arbre d'éléments :** Obligatoire. Contient des éléments dont un **élément racine;**
- Des **commentaires** et **instructions de traitement :** facultatifs.

NB : L'arbre est constitué d'éléments imbriqués les uns dans les autres ayant une relation parent-enfant et d'éléments adjacents.

Il existe deux représentations d'un document XML.

- ***Forme sérialisée*** : c'est la forme courante, où le contenu est marqué par des balises ;
- ***Forme arborescente*** : elle met en évidence la structure du document.

NB : Il est plus facile de raisonner sur la forme arborescente pour concevoir des traitements

Présentation de XML

Document sous forme sérialisée

```
<!-- Déclaration XML -->                                  }
<?xml version="1.0" encoding="ISO-8859-1" ?>              }
<!-- Instruction de traitement -->                        }
<?xml-stylesheet type="text/xsl" href="biblio.xsl"?>      }  Prologue
<!-- DTD -->                                              }
<!DOCTYPE Personne SYSTEM "Personne.dtd" [] >             }
<!--Début du document-->
<!-- Elément racine -->                                   }
<Personne>                                                }
        <Nom>Toto</Nom>                                   }  Arbre d'éléments
        <Prenom>Tutu</Prenom>                             }
</Personne>                                               }
<!--Fin du document-->
```

NB : Cette représentation permet le stockage et l'échange de documents.

Prologue

Déclaration XML

La déclaration **XML**, précède tout document XML.

Exemple **:**

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

- ***version*** : indique la version XML utilisée dans le document;
- ***encoding*** : indique le jeu de caractères utilisé dans le document (par défaut c'est **UTF-8**);
- ***standalone*** : indique si le document fait référence à une DTD externe (=**yes**) ou interne (=**no**). La valeur par défaut est **no**.

DTD ou Déclaration de type

Indiquer qu'un document est conforme à une DTD.

Exemple :

<!DOCTYPE Nom_Racine SYSTEM URI_DTD [Liste_Déclarations de la DTD interne]>

- ***NomRacine*** : nom de l'élément racine du document
- ***URI_DTD*** : la source extérieure contenant la DTD
- ***Liste_Déclarations*** : listes des déclarations

Présentation de XML

Arbre d'éléments

Un document est formé d'une hiérarchie (arbre) d'éléments et peut contenir :

- des éléments (et leurs attributs)
- des sections de texte
- des sections littérales

NB : Tout document comprend un et un seul élément racine

Les éléments

- Un nom d'élément ne contient pas de blanc, ni de caractère accentué
- les majuscules sont distinguées des minuscules
- un élément est de la forme: **<nom attr='valeur'> contenu </nom>**
- la forme abrégée pour les éléments sans contenu : **<Nom/>**
  - ***<nom>*** : balise d'ouverture
  - ***</nom>*** : balise de fermeture, obligatoire (sauf pour les éléments vides <nom/>)
  - ***contenu*** : contenu d'un élément qui peut être :
    - ✓ vide, texte, d'autres éléments, imbrication de texte et d'autres éléments
    - ✓ instructions de traitement, commentaires
  - ***attr='valeur'*** : ensemble éventuellement vide d'attributs
    - ✓ l'ordre des attributs n'est pas important
    - ✓ il doit toujours y avoir une valeur, encadrée par des apostrophes simples ou doubles
    - ✓ il ne peut pas y avoir deux attributs avec le même nom dans un élément.

Les sections littérales OU sections CDATA

Permettent d'inclure des caractères qui sont recopiés à l'identique, comme'<','>',ou'&'.

Exemple:

```
<?xml version='1.0'?>
<PROGRAMME>
        if((i<5)&&(j>6)) printf("error");
</PROGRAMME>
**********
```

} KO

```
<?xml  version='1.0'?>
<PROGRAMME>
        <![CDATA[if((i<5)&&(j>6))printf("error");]]>
</PROGRAMME>
```

} OK

Instruction de traitement

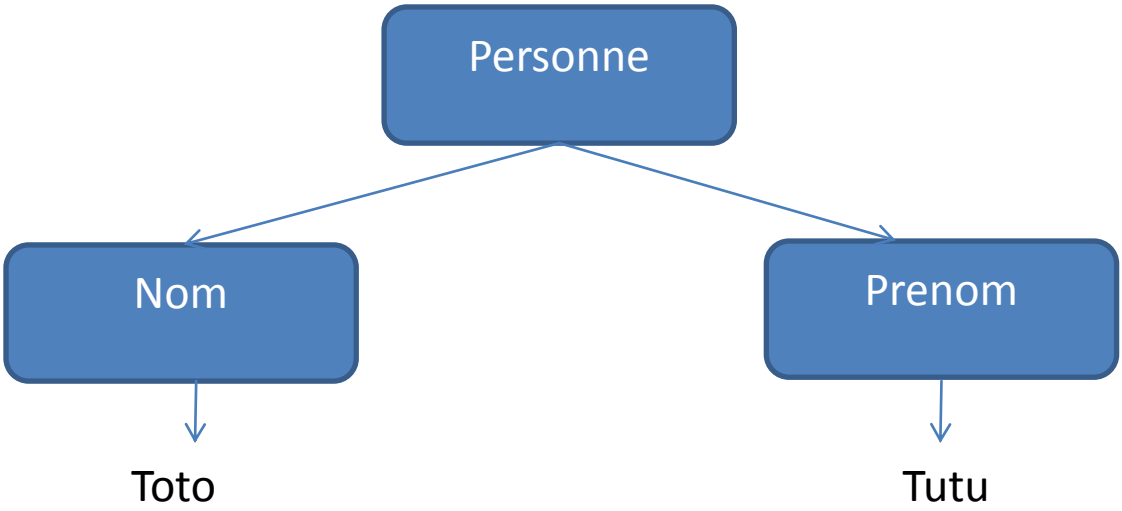
Elles sont destinées aux applications qui traitent les documents XML. On peut les utiliser dans n'importe quelle partie du document cependant elles sont en général utilisées dans le prologue.

Exemple :

```
<?xml-stylesheet type="text/xsl" href="biblio.xsl"?>
```

Document sous forme arborescente

La structure des arbres XML est définie par le Document Object Model (DOM)



Passage à la représentation DOM

Le document sérialisé est analysé, et une représentation arborescente est créée:

- le nœud racine est de type Document ;
- les catégories syntaxiques (commentaires, balises, texte) se traduisent par différents types de nœuds (Comment, Element, Text) ;
- les nœuds constituent un arbre qui reflète l'imbrication des éléments dans la forme sérialisée.

Contenu d'un nœud

- certains nœuds ont un nom (éléments), d'autres pas (texte, commentaires);
- Certains nœuds ont une valeur (attributs, texte), d'autres pas (éléments);
- Certains nœuds ont un contenu (éléments), d'autres pas(texte, attributs);
- Un contenu est un sous-arbre, une valeur est une chaîne de caractères sans balisage.