

Extensible Markup Language (XML)

Ahmad Mazyad

LISIC - Université du Littoral Côte d'Opale

2018

- 1 Support de Cours
- 2 Introduction
- 3 Syntaxe de XML
- 4 Espaces de noms (Namespaces)
- 5 Validation
- 6 Processing
- 7 Databases
- 8 Programming
- 9 Communication
- 10 Display

- Beginning XML 5 edition (JOE FAWCETT, ...)
Un très bon cours pour commencer mais en anglais
- L'essentiel de XML: Cours XML (Olivier Carton)
Support du cours XML en M2 Pro à l'Université Paris Diderot
<https://www.irif.fr/~carton/Enseignement/XML/Cours/support.pdf>
- Structurez vos données avec XML
Un cours en ligne, simple, facile à suivre, mais ...
<https://openclassrooms.com/courses/structurez-vos-donnees-avec-xml/qu-est-ce-que-le-xml>
- Attributes or Elements
<https://www.ibm.com/developerworks/library/x-eleatt/>
- slides: goo.gl/yPWqm6

What Is XML?

- Extensible Markup Language (Langage de balisage generique).
- Décrire et structurer des données selon certaines contraintes définies.

History

- SGML (Standard Generalized Markup Language)
- HTML - 1991
- XML - First draft en 1996
- XML - Première version publiée par W3C en 1998
- XHTML - Redéfinition du HTML 4.0 en 1999

XHTML

- HTML version XML (non utilisé depuis HTML 5)

XML vs HTML

- HTML décrit la présentation et XML décrit le contenu.
- HTML est une application de SGML et XML est “un sous-ensemble de SGML”

XML vs JSON

- JSON est un format de données.
- XML est une langage de balisage.

Applications XML

Configuration files, Web services, Web Content (XHTML), Document Management, Database Systems, Image Representatipon (SVG)

Syntaxe de XML

Example

```
1 <?xml version="1.0" encoding="UTF-8"?>①
2 <!-- Time-stamp: "bibliography.xml 3 Mar 2008" -->②
3 <!DOCTYPE bibliography SYSTEM "bibliography.dtd" >③
4 <bibliography>④
5     <book key="Michard01" lang="fr">⑤
6         <title>XML langage et applications</title>
7         <author>Alain Michard</author>
8         <year>2001</year>
9         <publisher>Eyrolles</publisher>
10        <isbn>2-212-09206-7</isbn>
11        <url>http://www.editions-eyrolles/livres/michard/</url>
12        <image href="michard.png" />⑥
13    </book>
14    <book key="Zeldman03" lang="en">
15        <title>Designing with web standards</title>
16        ...
17    </book>
18 </bibliography>⑦
```

Syntaxe de XML

Exemple

Explication

- ① Entête **XML** avec la version *1.0* et l'encodage *UTF-8* des caractères.
- ② Commentaire délimité par les chaînes de caractères `<!--` et `-->`.
- ③ Déclaration de **DTD** externe dans le fichier *bibliography.dtd*.
- ④ Balise **ouvrante** de l'élément racine *bibliography*
- ⑤ Balise ouvrante de l'élément *book* avec deux attributs de noms *key* et *lang* et de valeurs *Michard01* et *fr*
- ⑥ Balise **auto fermante** `<image />`
- ⑦ Balise **fermante** de l'élément racine *bibliography*

Syntaxe de XML

Prologue

```
1  Entête
2  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
3  Commentaire
4  <!-- Time-stamp: "bibliography.xml 3 Mar 2008" -->
5  DOCTYPE
6  <!DOCTYPE bibliography SYSTEM "bibliography.dtd" >
```

Entête

- Première partie d'un document XML.
- L'attribut *version* précise la version de XML utilisée (*1.0*, *1.1*).
(**Obligatoire**)
- *encoding* indique le jeu de caractères utilisé (**UTF-8**, US-ASCII, ISO-8859-1, et UTF-16).
- *standalone* permet d'indiquer si le document est autonome ou rattaché à un fichier DTD (**no**, **yes**).


```
1 <element [attribut="valeur" autre_attribut="valeur2"]>
2   [<element_auto_fermante [attribut="valuer"] />
3   <element_imbriueque>[data]</element_imbriueque>
4   data]
5 </element>
```

Élément

- Formé d'une balise ouvrante, d'un contenu et d'une balise fermante.
- Quand le contenu est vide, on peut utiliser la balise auto fermante `< balise/ >`.
- Un élément peut avoir zero, un ou plusieurs attributs uniques.

Imbrication des Eléments

`< balise1 >< balise2 >< /balise1 >< /balise2 >` (Invalide)

Tous les documents doivent avoir un seul élément racine.

```
1 <element [attribut="valeur" autre_attribut="valeur2"]>  
2   [<element_auto_fermante [attribut="valuer"] />  
3   <element_imbriueque>[data]</element_imbriueque>  
4   data]  
5 </element>
```

Attribut

- Un attribut est constitué d'un nom et d'une valeur séparés par un signe égal =.
- La valeur de l'attribut doit être entre guillemets simples ou doubles.
- Un attribut doit être unique pour un élément.

Attributs particuliers

- `xml:lang` décrit la langue du contenu de l'élément
`<p xml:lang="fr">Bonjour</p>`
- `xml:space` permet d'indiquer à une application le traitement des caractères d'espacement (**default** et `preserve`).
- `xml:base` permet de préciser l'URI de base d'un élément.
`<chapter xml:base="XML/chapter.html">`
- `xml:id` permet d'associer un identificateur à tout élément indépendamment de toute DTD ou de tout schéma.

Style de Nomenclature (Naming Style)

- Pascal-casing: Ceci met en majuscules les mots séparés, y compris le premier: `< MyElement / >`.
- Camel-casing: Similaire à Pascal sauf que la première lettre est en minuscule: `< myElement / >`.
- Mots séparés par un trait de soulignement: `< my_element / >`

Spécifications de Nomenclature

Règles à suivre pour nommer les éléments et les attributs.

- le nom d'un élément/Attribut commence avec `[a-zA-Z_]`.
- Les caractères suivants peuvent également être un tiret (-) ou un chiffre.
- Les noms étant sensibles à la casse, les balises de début et de fin doivent correspondre exactement.
- Ne pas utiliser XML au début d'un nom.

```
1 <conversionData>
2   1 kilometer &lt; 1 mile
3   1 pound &lt; 1 kilogram
4 </conversionData>
5 <conversionData><![CDATA[
6   1 kilometer < 1 mile
7   1 pound < 1 kilogram
8 ]]></conversionData>
```

Sections Littérales/CDATA

Permet d'utiliser les caractères spéciaux (<) directement au lieu de des Entités prédéfinies (<).

Quand utiliser les éléments et quand les attributs?

```
1 <applicationUsers>
2   <user firstName="Joe" lastName="Fawcett" />
3   <user firstName="Danny" lastName="Ayers" />
4 </applicationUsers>

5
6 <applicationUsers>
7   <user>
8     <firstName>Joe</firstName>
9     <lastName>Fawcett</lastName>
10  </user>
11  <user>
12    <firstName>Danny</firstName>
13    <lastName>Ayers</lastName>
14  </user>
15 </applicationUsers>
```

Corps du Document

XInclude

```
1 <book xmlns:xi="http://www.w3.org/2001/XInclude">
2   ...
3   <!-- Inclusion des differents chapitres -->
4     <xi:include href="introduction.xml" parse="xml"/>
5     <xi:include href="Syntax/chapter.xml" parse="xml"/>
6   ...
7 </book>
8 -----
9 <xi:include xmlns:xi="http://www.w3.org/2001/XInclude"
10 href="file.php?id=1&c=generatexmlfile" />
```

XInclude

répartir un gros document en plusieurs fichiers afin d'en rendre la gestion plus aisée.

Exercice 1

Il vous est demandé de réaliser un format XML permettant de stocker des recettes de cuisine et gérant également les stocks des ingrédients que vous possédez. Chaque recette de cuisine a un nom, une description, la durée de préparation et la durée de cuisson, le nombre de calories par personne, le nombre de parts et le niveau de difficulté : difficile, moyen ou facile. Pour chaque recette vous voulez savoir quels sont les ingrédients nécessaires et la quantité associée à chaque ingrédient. Pour chaque ingrédient vous avez son nom et le nombre de calories pour 100 grammes de cet ingrédient. Un même ingrédient peut avoir plusieurs conditionnements, par exemple, l'ingrédient farine peut être stocké sous forme d'un paquet de 1 kg ou de 500 g. Pour gérer les stocks des ingrédients, le lieu de stockage de produits(conditionnement) est mémorisé.

Question: Proposer un format XML.

Exercice 2

L'objectif de l'exercice est de proposer un format XML permettant de représenter les étapes ainsi que les informations de chaque étape de la fabrication du champagne.

- Le pressurage consiste à presser manuellement le raisin pour en faire éclater les baies.
- La fermentation est la phase de mise en cuve du jus de raisin. Celle-ci dure quinze jours environ. Conservée à une température constante de 18-20 °C, cette 1ère fermentation dite "alcoolique" active les levures naturelles présentes dans le jus de raisin et transforme les sucres en un mélange d'alcool et de gaz carbonique.
- La clarification débarrassera le vin des levures ou autres particules solides qui altèrent sa saveur.
- Le vigneron réalise l'assemblage en mélangeant des vins "tranquilles" (non effervescents) issus de différentes récoltes pour une saveur finale constante. Cette étape constitue la véritable "signature" d'une cuvée de champagne.
- Le vin obtenu est mis en bouteille, additionné de sucre et levures. Cette seconde fermentation le transforme en vin effervescent. L'étape du tirage permet la "prise de mousse", la bouteille est bouchée d'un "bidule" (capsule).
- Le vin reposera 15 mois pour la fabrication d'un champagne brut et 36 mois pour obtenir un Millésime.
- Le remuage, manuel ou mécanique, consiste à faire tourner la bouteille de gauche à droite, puis à lui mettre la tête en bas, pour faciliter l'accumulation des dépôts organiques.
- Ceux-ci seront gelés puis évacués naturellement lors du dégorgement effectué par le col de la bouteille plongé dans un liquide à -25°C. La bouteille est alors prête à recevoir le bouchon de liège, la capsule, le muselet, l'étiquette et la collerette qui finalisent la fabrication du champagne.

Question: Proposer un format xml.

Espaces de noms (Namespaces)

Espaces de noms?

Les espaces de noms sont un moyen de regrouper les éléments et les attributs sous un en-tête commun afin de les différencier des éléments portant le même nom.

Choisir un Nom?

N'importe quelle chaîne de caractères peut être utilisée.

W3C recommande d'utiliser les URIs (URL, URN).

e.g. <http://univ-littoral.fr/namespaces/m1/xml>

En Pratique

Les espaces de noms sont surtout utilisés dans ces cas:

- XML Schemas: définir la structure d'un document
- Combinaison des documents de plusieurs sources.
- Versioning

Espaces de noms

Example

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <book version="5.0" xml:lang="fr"
   xmlns="http://docbook.org/ns/docbook"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:xi="http://www.w3.org/2001/XInclude">
3   <title>Langages formels</title>
4   <!-- MEtadonnEes -->
5   <metadata>
6     <dc:title>Langages formels</dc:title>
7     <dc:creator>Olivier Carton</dc:creator>
8     <dc:date>2008-10-01</dc:date>
9   </metadata>
10  <!-- Import des chapitres avec XInclude -->
11  <xi:include href="introduction.xml" parse="xml" />
12  <xi:include href="chapter1.xml" parse="xml" />
13 </book>
```

Déclarer un espace de noms

```
1 <!-- Declare a namespace -->
2 <td:elements
    xmlns:td="http://univ-littoral.fr/namespaces/m1/td">
3   <elt attr1="..." attr2="..." />
4   <elt attr1="..." attr2="..." />
5 </td:elements>
```

```
1 <!-- Multiple namespaces -->
2 <td:elements
    xmlns:td="http://univ-littoral.fr/namespaces/m1/td"
3   xmlns:tp="http://univ-littoral.fr/namespaces/m1/tp">
4   <tp:elt attr1="..." attr2="..." />
5   <tp:elt attr1="..." attr2="..." />
6 </td:elements>
```

```
1 <!-- Default namespaces -->
2 <elements
   xmlns="http://univ-littoral.fr/namespaces/m1/td">
3   <elt attr1="..." attr2="..." />
4   <elt attr1="..." attr2="..." />
5 </elements>
```

Attention

Une fois un élément/attribut est associé à un espace de noms, son nom complet devient {URI}nom_element (Convention mais utilisée par les différents analyseurs/parsers)

.e.g le nom de l'élément "elt" ci-dessus associé à l'espace de noms par défaut est:

{http://univ-littoral.fr/namespaces/m1/td}elt

Conclusion

- Un moyen de regrouper les éléments et les attributs sous un en-tête commun.
- On peut utilisé n'importe quel nom, mais c'est conseillé d'utiliser un URI.
- On peut choisir entre un espace de noms par défaut dont les éléments sont associés automatiquement ou une declaration d'espace de noms préfixés.
- Le préfixe peut utiliser n'importe quelle chaîne de caractères contenant pas ":". Pour une question de praticité, on utilise une chaine simple et courte.

Exercice

Exercice 3 - Espaces de noms

- Déclarer un espace de noms par défaut dans l'élément racine du document XML de l'exercice 1 (Recettes).
- Déclarer un espace de noms préfixé (prefix et url au choix), et associé certains éléments à l'espace déclaré.

Document Type Definitions (DTD)

Introduction

DTD?

Le rôle d'une DTD (Document Type Definition) est de *définir* précisément la *structure d'un document*. Il s'agit d'un certain nombre de contraintes que doit respecter un document pour être valide. Ces contraintes spécifient:

- Les éléments qui peuvent apparaître dans le contenu d'un élément
- L'ordre éventuel de ces éléments.
- La présence de texte brut.
- Les attributs autorisés et les attributs obligatoires pour chaque éléments.

Pourquoi utiliser une DTD?

Exemple d'Utilisation

1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```


Document Type Definitions (DTD)

Example

```
1 <!ELEMENT bibliography (book)+> ①
2 <!ELEMENT book (title, author, year, publisher, isbn,
   url?)> ②
3 <!ATTLIST book key NMTOKEN #REQUIRED> ③
4 <!ATTLIST book lang (fr | en) #REQUIRED> ④
5 <!ELEMENT title (#PCDATA)> ⑤
6 <!ELEMENT author (#PCDATA)>
7 <!ELEMENT year (#PCDATA)>
8 <!ELEMENT publisher (#PCDATA)>
9 <!ELEMENT isbn (#PCDATA)>
10 <!ELEMENT url (#PCDATA)>
```

① Déclaration de l'élément bibliography devant contenir une suite non vide d'éléments book.

② Déclaration de l'élément book devant contenir les éléments title, author, ..., isbn et url.

③④ Déclarations des attributs obligatoires key et lang de l'élément book.

⑤ Déclaration de l'élément title devant contenir uniquement du texte.

Document Type Definitions (DTD)

Déclaration Interne

Déclaration Interne

Lorsque la DTD est incluse dans le document, sa déclaration prend la forme suivante: `<!DOCTYPE root-element [declarations]>`

Utilisation utile en phase de développement.

```
1 <?xml version="1.0"?>
2 <!DOCTYPE name [
3     <!ELEMENT name (first, middle, last)>
4     <!ELEMENT first (#PCDATA)>
5     <!ELEMENT middle (#PCDATA)>
6     <!ELEMENT last (#PCDATA)>
7 ]>
8 <name>
9     <first>Joseph</first>
10    <middle>John</middle>
11    <last>Fawcett</last>
12 </name>
```

Document Type Definitions (DTD)

Déclaration Externe

Déclaration Externe

Lorsque la DTD est dans un fichier externe, on donne l'adresse du DTD.

Privée: `<!DOCTYPE root-element SYSTEM "url" [declarations]>`

Public: `<!DOCTYPE root-element PUBLIC "fpi" "url" [declarations]>`

Formal Public Identifier (fpi)

format: `type//owner//desc//lang`

`-//W3C//DTD XHTML 1.0 Strict//EN`

`ISO//IEC 10179:1996//DTD DSSSL Architecture//EN`

```
1 <?xml version="1.0"?>
2 <!DOCTYPE name SYSTEM "name.dtd" []>
3 <name>
4   <first>Joseph</first>
5   ...
6 </name>
```

Document Type Definitions (DTD)

Contenu de la DTD

Contenu de la DTD

Une DTD est généralement constitué de trois type de déclarations:

- Déclaration des Eléments. **ELEMENT**
- Déclaration des Attributs. **ATTLIST**
- Déclaration des Entités. **ENTITY**

Exemples

- element `<message>Bonjour</message>`
- attribut ``
- entités: `<`; `>`; `&`; `"`; `'`;

Document Type Definitions (DTD)

Déclaration des Elements

Déclaration des Elements

`<!ELEMENT element-name category>` ou
`<!ELEMENT element-name (element-content)>`

Example

Un seul occurrence : `<!ELEMENT contact (name)>`
plusieurs occurrence : `<!ELEMENT contact (name+)>`
Sequenne: `<!ELEMENT name (first, middle, last)>`
Choix: `<!ELEMENT location (address | GPS)>`
Combinaison: `<!ELEMENT location (address | (latitude, longitude))>`
Text: `<!ELEMENT first (#PCDATA)>`
Mixte: `<!ELEMENT description (#PCDATA | title | detail)*>`
Vide: `<!ELEMENT br EMPTY>`
Any: `<!ELEMENT description ANY>`

Cardinalité

+ Indique qu'un élément peut apparaître une ou plusieurs fois.
*: zero ou plus, ?: zero ou un, none: un

Document Type Definitions (DTD)

Déclaration des Attributs

Déclaration des Attributs

```
<!ATTLIST element—name attribute—name attribute—type  
attribute—value>
```

Type (attribute-type)

- CDATA: text (default).
- ID: unique.
- IDREF: reference par ID.
- IDREFS: liste des valeurs d'IDREF séparés par des espaces.
- ENTITY: entité externe.
- ENTITIES: liste ENTITY.
- NMTOKEN: token name.
- NMTOKENS: liste NMTOKEN
- (en1—en2—..) Liste énumérée des valeurs possibles.

Document Type Definitions (DTD)

Déclaration des Attributs

Déclaration des Attributs

`<!ATTLIST element-name attribute-name attribute-type attribute-value>`

Valeur (attribute-value)

- valeur par défaut `<!ATTLIST phone kind (Home | Work)"Home">`
- valeur fixée `<!ATTLIST contacts version CDATA #FIXED "1.0">`
- requis `<!ATTLIST phone kind (Home | Work)#REQUIRED>`
- optionel `<!ATTLIST knows contacts IDREFS #IMPLIED>`

Document Type Definitions (DTD)

Déclaration des Entités

Entités

Les entités sont utilisés dans les documents XML pour faire référence à de simple caractères (Entités prédéfinies), des sections de texte, d'autre balises, et même des fichiers externes. On peut séparer les entités en quatre types principaux.

- Entités prédéfinies < (&), >(>), & (<), ' ('), " (")
<description>Author & programmer</description>
- Entités de Caractères: &#unicode;
<price>30 €</price>
- Entités Générales
<!ENTITY name "fragment">
<!ENTITY titre "Cours XML">
<title>&titre;</title>
- Entités de Paramètres (Uniquement dans DTD):
<!ENTITY % name "fragment">
<!ENTITY % DefaultPhoneKind ""Home"">
<!ATTLIST phone kind (Home | Work)"%DefaultPhoneKind;">

Document Type Definitions (DTD)

Exercice

Exercices

- Exo 4 - Créer un fichier DTD pour valider le document XML de l'exercice 1 (Recettes).
- Exo 5 - Créer un fichier DTD pour valider le document XML de l'exercice 2 (Fabrication du Champagne).

XML and Databases

LINQ to XML

Scalable Vector Graphics (SVG)

