

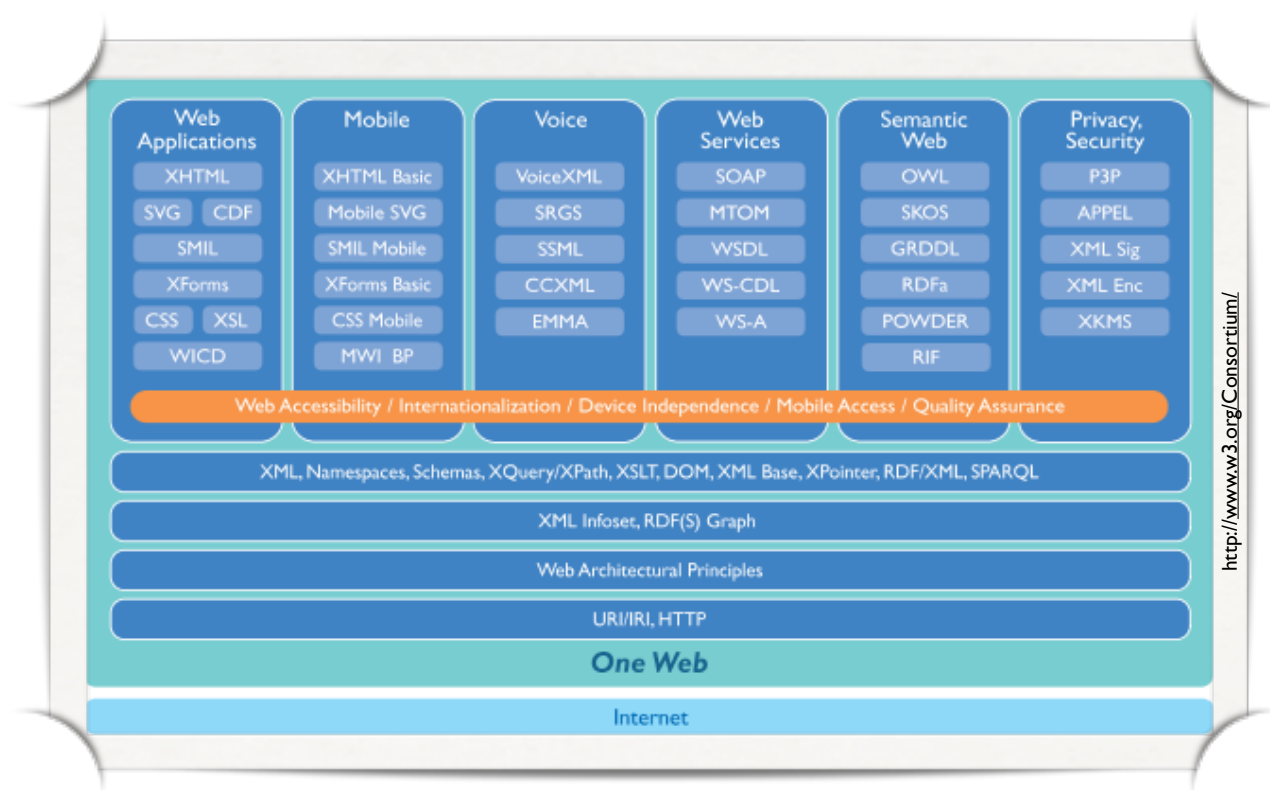
Fiche XML n°1

Syntaxe

Cours

1. XML, un «méta-langage», un standard ?

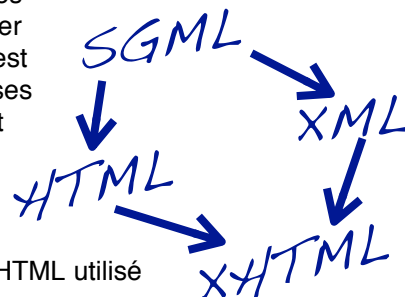
XML (eXtensible Markup Language¹) est un standard informatique du W3C², très lié à Internet (au Web en particulier), créé en 1998. Le W3C propose de nombreux langages basés sur la syntaxe XML. La figure ci-dessous en présente un certain nombre dans plusieurs domaines (applications Web, informatique mobiles, voix, services Web, etc.)



XML est un «méta-langage» permettant de construire des langages spécifiques pour représenter des documents ou pour structurer (organiser) des données, sous forme d'éléments balisés imbriqués. C'est donc un «langage», ou plutôt un «méta-langage», utilisant des balises (repères s'utilisant par paires à la manière des parenthèses), permettant de décrire des données ou des textes.

XML est une simplification d'un langage appelé SGML et utilisé pour structurer des documentations techniques dans de très grosses entreprises. Sa structure, quoique plus stricte, s'approche du langage HTML utilisé pour construire les pages Web.

XML, en plus d'être structurant, permet aussi de spécifier les normes utilisées (langue, codage des caractères, etc.).



¹ Extensible Markup Language (XML) 1.0. W3C Recommendation. REC-xml-19980210. 10 February 1998, mise à jours le 26 novembre 2008 par REC-xml-20081126 ; <http://www.w3.org/TR/REC-XML>

² World Wide Web Consortium, consortium de sociétés et d'universités (365 membres) créé en octobre 1994 dont l'objectif est de proposer des standard pour le Web, principalement basés sur l'utilisation de XML (mais pas seulement) <http://www.w3c.org>

Structurer un document avec XML apporte certains avantages :

- c'est un standard indépendant d'une application propriétaire ; la pérennité des informations est ainsi augmentée ;
- le document est «lisible» directement avec un éditeur de texte (Notepad par exemple), sans passer par des applications lourdes ;
- le document est indépendant d'un environnement particulier (machine, système d'exploitation, application, etc.).

XML est un ensemble de règles de syntaxe et de «grammaire» permettant de construire un langage particulier grâce à des schémas de document (en définissant de nouvelles règles de grammaire et en spécifiant le vocabulaire utilisable). C'est un langage permettant de construire des langages spécifiques : il joue donc le rôle de «méta-langage».

De plus, il existe des outils permettant d'explorer un document construit avec XML (XPath, XQuery) ou de transformer le document (XQuery, XSLT).

2. Règles de syntaxe élémentaires

Définition : une balise XML (ou tag) est une étiquette, un marqueur, permettant de localiser une information dans un texte et de lui apporter éventuellement une sémantique.

Les balises sont utilisées toujours par paire, comme les parenthèses : à chaque balise «ouvrante» est associée une balise «fermante». Entre les deux, une information associée à la balise (on parle alors du contenu de la balise).

Dans l'exemple suivant, les balises sont en caractères gras : «**<nom-propre>**Dupont**</nom-propre>**». La balise ouvrante «**<nom-propre>**» et la balise fermante «**</nom-propre>**» identifient la chaîne de caractères «Dupont», le contenu. La différence entre balise ouvrante et balise fermante est uniquement la présence d'un «/» au début de la balise fermante.

Définition : un élément XML est une information encadrée par une balise ouvrante et une balise fermante.

Pour reprendre l'exemple précédent, «**<nom-propre>**Dupont**</nom-propre>**» est un élément XML.

Une «information» peut être soit du texte, soit des éléments XML, soit une alternance de textes et d'éléments XML. Par exemple, l'exemple ci-dessous propose un élément XML comportant des éléments XML.

```
<identité>
  <nom>Dupont</nom>
  <prénom>Marc</prénom>
</identité>
```

Attention, il ne faut pas croiser les balises pour des éléments différents. Par exemple «**<nom>**Dupont**<prénom>**<**</nom>**Marc**</prénom>**» n'est pas correct. De plus, il existe une exception à la règle de la parité des balises. En effet, si l'élément ne contient rien, il existe une écriture abrégée équivalente avec une balise qui est en même temps ouvrante et fermante.

Par exemple, «**<téléphone>**<**</téléphone>**» s'écrit aussi «**<téléphone/>**».

Le nom d'une balise respecte la règle de construction des noms XML. Cette règle indique que n'importe quel caractère alphanumérique et/ou graphique est autorisé. De même, certains caractères de ponctuation sont utilisables comme '_', '-', ':' et '.'. Cependant, quelques contraintes sont posées :

- un nom XML commence par une lettre ou '_' (pas de chiffres, '-', ':' ou '.') ;
- il est sensible à la casse ("Nom" ≠ "nom") ;
- il ne commence pas par "xml" (réservé aux noms pour XML... seuls deux sont actuellement identifiés «xml:lang» et «xml:preserve») ;
- il n'y a pas de limite de longueur ;
- '.' est autorisé, mais il est réservé à la manipulation des espaces de noms (abordés dans une autre fiche) ;
- les autres caractères de ponctuation sont interdits.

Voici quelques exemples de noms de balise qui ne sont pas valides comme noms XML :

- `<Num_d'identification>544-6855-545 </Num_d'identification>` : le caractère «'» n'est pas autorisé dans les noms XML ;
- `<mois/jours/annee>12-31-2004 </mois/jours/annee>` : là aussi, «/» n'est pas autorisé dans les noms XML ;
- `<nom famille>Dupont</nom famille>` : ici, deux erreurs sont possibles : soit le nom XML comporte un espace, caractère interdit, soit «famille» est un attribut (voir plus loin), il faut alors une valeur de la forme «='quelquechose'» ;
- `<3-tiers>Client/Middle/Server</3-tiers>` : un nom XML ne peut pas commencer par un chiffre.

Définition : un attribut XML est un couple (nom, valeur) qui apparaît à l'intérieur d'une balise d'ouverture, après son nom et qui qualifie un élément. Il apparaît sous la forme «nom="valeur"» ou bien «nom='valeur'». Le nom d'un attribut respecte la syntaxe d'un nom XML.

Un attribut de nom donné ne peut être présent qu'une seule fois dans une balise. Par contre, il peut y avoir plusieurs attributs dans une balise et leur ordre n'est pas significatif.

Par exemple, l'élément XML «`<date iso='2009-09-09'>le neuf septembre 2009</date>`» contient une balise «date» pour laquelle apparaît un attribut de nom «iso» et de valeur «2009-09-09» qui correspond à la forme ISO8601³ de la date indiquée en toutes lettres dans le contenu de la balise.

Définition : un document XML est un document composé :

- d'une ligne d'entête qui doit se trouver sur le premier caractère de la première ligne ; elle commence par «`<?xml`» et termine par «`?>`» et contient au moins un attribut «version» (valeur «1.0» ou «1.1») qui référence la version du standard XML utilisé par le fichier ;
- d'une ou plusieurs lignes d'entête complémentaires facultatives qui seront étudiées dans d'autres fiches ;
- d'un seul et unique élément XML appelé *élément racine* du document.

L'exemple ci-dessous présente un document XML décrivant un annuaire téléphonique simpliste. L'élément racine est délimité par les balises «`<annuaire>`» et «`</annuaire>`».

```
<?xml version="1.0" ?>
<annuaire>
  <personne>
    <nom>Dupont</nom>
    <prenom>Paul</prenom>
    <telephone>02 54 55 98 66</telephone>
  </personne>
  <personne>
    <nom>Alain</nom>
    <prenom>Adrien</prenom>
    <prenom>Paul</prenom>
    <telephone>02 40 40 36 18</telephone>
  </personne>
  ...
</annuaire>
```

D'autres structures peuvent aussi être présentes dans un document XML. Il est possible de trouver, par exemple, des «sections CDATA» et des commentaires.

Définition : une section CDATA est une partie de fichier XML qui ne sera pas analysée. Elle commence par «`<![CDATA[`» et termine par «`]]>`». Elle contient toute suite de caractères, même si la suite ne respecte pas les règles de syntaxe XML (sauf bien sûr la suite de caractères «`]]>`»).

³ l'ISO8601 est un standard international pour la représentation des dates et des heures. http://fr.wikipedia.org/wiki/ISO_8601

Définition : un *commentaire XML* est une chaîne de caractères de longueur quelconque, comprise entre '`<!--`' et '`-->`'. Cette suite de caractères est ignorée lors de l'analyse du document. Le commentaire peut être positionné à tout endroit en XML sauf à l'intérieur d'une balise et devant la première ligne d'entête.

Par exemple, le document suivant comporte deux commentaires et une section CDATA :

```
<?xml version="1.0" encoding="UTF-8"?>
<racine> <!-- Début d'une section CDATA -->
<![CDATA[
Ceci <est> un essai & !
]]> <!-- Fin de la section CDATA -->
</racine>
```

Définition : un *document XML bien formé* est un document respectant les règles XML énoncées précédemment, donc pour mémoire :

- les éléments correctement imbriqués et comportant des balises ouvrantes et fermantes ;
- l'élément racine est unique ;
- les noms d'éléments, d'attributs... sont des noms XML ;
- les attributs sont entre guillemets ou apostrophes ;
- il n'y a pas deux attributs de même nom dans une balise ;
- les commentaires sont en dehors des balises ;
- d'autres règles existent sur des notions qui seront abordées dans d'autres fiches.

3. Conclusion

Pour conclure, XML est un (méta-)langage simple et souple, adapté aux formats internationaux (codage des caractères, etc.). Comme méta-langage, il est le point de départ de différents langages (ou dialectes) dans un grand nombre de domaines comme :

- les bases de données ;
- la gestion documentaire ;
- l'interface Homme-Machine ;
- le multimédia ;
- la représentation des connaissances ;
- la sécurisation...

Remarque : nous le verrons plus en détail dans une prochaine fiche, mais certains caractères ne sont pas acceptés directement. Par exemple, le symbole «`<`» est utilisé pour la construction d'une balise. Aussi, si le contenu d'un élément contient le symbole «`<`» il doit être remplacé par un codage «`<`». Il en est ainsi pour d'autres caractères comme le «`&`» qui devient «`&`», etc.

Pour des objectifs plus orientés vers le développement informatique, une autre structure existe en XML.

Définition : une *instruction de traitement* est une structure qui contient une chaîne qui ne sera pas traitée comme du XML mais qui est destinée à être traitée par une application. Elle débute par «`<?nom_cible`» et termine par «`?>`». Entre ces deux limites, se trouve un code lié à «`nom_cible`» (qui est un nom XML), le code d'un langage informatique ou d'une application, souvent des attributs destinés à une application tierce.

L'instruction de traitement se place partout dans le document, sauf à l'intérieur d'une balise et devant la première ligne d'entête. Un exemple d'instruction de traitement souvent utilisé est la référence à une feuille de style XSL :

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

NB : une feuille de style est un document, dans un langage spécifique, décrivant les règles de présentation attachées au document XML pour une représentation Web. Les principaux langages sont CSS et XSL.

Exercices

Exercice 1. Annuaire

Dans l'exemple ci-contre, relever les nombreuses erreurs qui font que ce document n'est pas bien formé. Proposer des solutions pour les corriger.

```

01 : <?xml version="1.0" ?>
02 : <!-- Annuaire d'illustration -->
03 : <annuaire>
04 :     <personne <!-- une 1ere personne --> >
05 :         <nom>Dupont</nom>
06 :         <prénom>Paul <!-- ou Polo --> </prénom>
07 :         <date format="ISO" format="fr-fr">1998-07-23
08 :         <telephone/>
09 :     </personne>
10 :     <personne>
11 :         <!-- une 2nd personne --><nom>Alain</nom>
12 :         <prenom courant='Vrai'>Adrien</prenom>
13 :         <date format=ISO>1969-06-04</date>
14 :         <telephone>
15 :             <indicatif tel> 02 </indicatif tel>
16 :             <num#tel> 40 40 36 18 </num#tel>
17 :         </personne>
18 :     </telephone>
19 : </annuaire>
20 : <annuaire>
21 : <personne>
22 :     <nom>DUPONT</nom>
23 :     <prenom>MARC</prenom>
24 :     <telephone/>
25 : </personne>
26 : </annuaire>

```

Exercice 2. Erreurs

Les documents ci-dessous sont-ils des documents XML bien formés ? Si ce n'est pas le cas, indiquer les solutions possibles pour les corriger.

<pre> 01 : <?xml version="1.0" ?> 02 : <racine> 01 : <?xml version="1.0" ?> 02 : <début_racine></fin_racine> </pre>	<pre> 01 : <?xml ?> 02 : <racine>Quand on a a<b alors il faut lancer la procédure b()... 03 : </racine> </pre>
<pre> 01 : <project name="Dinosys" default="compile" basedir="."> 02 : <property name="build" location="WEB-INF/classes" /> 03 : <property name="src" location="WEB-INF/src"/> 04 : <target name="compile" 05 : description="compile the source " > 06 : <!-- Compile the java code from \${src} 07 : into \${build} --> 08 : <javac classpath="\${classpath}..." srcdir="\${src}" 09 : destdir="\${build}"/> 10 : </target> 11 : </project> </pre>	<pre> 01 : <?xml version="1.0" 02 : encoding="iso-8559-1" ?> 03 : <racine> 04 : Quand on a a<b alors il faut lancer la procédure b()... 05 : </racines> </pre>
<pre> 01 : <?xml version = '1.1'?> 02 : <records> 03 : <cd> 04 : <title>None too soon</title> 05 : <artist>Allan Holdsworth</artiste> 06 : <country>UK</country> 07 : <company>Polidor</company> 08 : <year>1996</cd> 09 : </year> 10 : <cd> 11 : <title>All night wrong</title> 12 : <artiste>Allan Holdsworth</artiste> 13 : <country>UK</country> 14 : <company>Polidor</company> 15 : <year>2002</year> 16 : </cd> 17 : </record> </pre>	<pre> 01 : <?xml encoding="iso-8559-1" ?> 02 : <racine>Quand on a a<b alors il faut lancer la procédure b suivante : 03 : function b(\$x) { 04 : x=24; 05 : for(\$i=1;\$i<=10;\$i=\$i+1) { 06 : x=x+1;} 07 : return x; 08 : } 09 : </racine> </pre>

Nantilus : Portail documentaire de l'Université de Nantes

NANTILUS
Portail documentaire de l'université de Nantes

COMPTE LECTEUR | MOT DE PASSE OUBLIÉ ?

Vous êtes étudiant ou personnel de l'université
Vous ne faites pas partie de l'université

Rechercher

- Catalogue
- Revues
- Thèses et mémoires
- Ressources électroniques
- Recherche multi-ressources
- SUDOC
- Historique des recherches

Consulter

- Dernières acquisitions
- Guide des ressources
- Guide des services
- Offre de formation

S'informer

- Présentation de la bibliothèque universitaire
- Charte documentaire
- Bibliothèques associées et partenaires
- Bibliothèques de l'agglomération nantaise

Accueil > Catalogue - Recherche avancée > Liste des résultats

Résultats de votre recherche

Modifier ma recherche ?

Rappel de la recherche
(Limiter par : BU Sciences) ET (Titre:XML)

56 résultats

Actions sur la sélection :

Ressources

Trier par : Aucune OK

Limiter par : BU Sciences

Tout cocher / Tout décocher

Catalogue des bibliothèques de l'Université de Nantes - (Résultats 1 à 10 sur 56)

- ☐ **Programmer pour Internet : notions fondamentales et mise en pratique (tcp/ip, http, html, java, php, MySQL, XML...)**
Cozette, Philippe -- 19... -- Ingénieur informaticien
Éd. ENI -- cop. 2007
- ☐ **XML : informatique : synthèse de cours & exercices corrigés**
Chagnon, Gilles
Pearson Education -- Impr. 2007
- ☐ **XML : cours et exercices : modélisation, schéma, design patterns, XSLT, XPath, SOAP, XQuery, XSL-FO, SVG**
Brillant, Alexandre -- 1973?
Eyrolles -- DL 2007
- ☐ **Beginning XML databases**
Powell, Gavin
Wiley -- cop. 2007
- ☐ **Modélisation XML**
Lonjon, Antoine
Eyrolles -- DL 2006
- ☐ **XML Schema et XML Infoset : pour les services web**
Lamareille, Vincent
Cepaduès-Éd. -- DL 2006

Exercice 3. BU

En vous aidant des deux écrans disponibles issus du logiciel de consultation de la BU de Nantes⁴, proposez un document XML structurant toutes les données qui seraient nécessaires à la production de telles pages. Vous intégrerez environ trois ouvrages à votre document. Le second écran est obtenu après avoir choisi un titre particulier dans le tableau du premier écran. Expliquer les choix et détailler les hypothèses posées.



Cette création est mise à disposition selon le Contrat Paternité-Pas d'Utilisation Commerciale-Pas de Modification 2.0 France disponible en ligne <http://creativecommons.org/licenses/by-nc-nd/2.0/fr/> ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

⁴ <http://www.bu.univ-nantes.fr/>

Nantilus : Portail documentaire de l'Université de Nantes

NANTILUS
Portail documentaire de l'université de Nantes

COMPTE LECTEUR | MOT DE PASSE OUBLIÉ ?

Vous êtes étudiant ou personnel de l'université
Vous ne faites pas partie de l'université

Rechercher

- Catalogue
- Revues
- Thèses et mémoires
- Ressources électroniques
- Recherche multi-ressources
- SUDOC
- Historique des recherches

Consulter

- Dernières acquisitions
- Guide des ressources
- Guide des services
- Offre de formation

S'informer

- Présentation de la bibliothèque universitaire
- Charte documentaire
- Bibliothèques associées et partenaires
- Bibliothèques de l'agglomération nantaise

Accueil > Catalogue - Recherche avancée > Liste des résultats > Notice

Résultats de votre recherche

Modifier ma recherche < Retour à la liste de résultats ?

Rappel de la recherche
(Limiter par :) ET (Titre:XML)

Actions :

Notice 5 sur 56

Modélisation XML
Par Lonjon, Antoine, Thomasson, Jean-Jacques, Maesano, Libero

Auteurs : Lonjon, Antoine
Thomasson, Jean-Jacques
Maesano, Libero

Adresse : Paris, Eyrolles -- DL 2006

Description : 1 vol. (XXVII-499 p.) : ill. ; 23 cm

Collection : Architecte logiciel

Notes : Bibliogr. p. 453-454. Glossaire. Index

Isbn : 2-212-11521-0 br. 42 EUR

Sujets : Logiciels -- Développement
XML (langage de balisage)

Titre et zones de contenu : Modélisation XML, Texte imprimé -- Antoine Lonjon, Jean-Jacques Thomasson ; ouvrage dirigé par Libero Maesano

Données spécifiques d'exemplaires

Bibliothèque	Localisation	Type de prêt	Cote	Année	Statut	Date de retour
BU SCIENCES	1er étage, salle 12	EMPRUNTABLE	681.321 XML-LON		Disponible	
BU TECHNOLOGIES	ouvrages généralités, informatique	EMPRUNTABLE	005.133 XML-LON		Disponible	
BU TECHNOLOGIES	ouvrages généralités, informatique	EMPRUNTABLE	005.133 XML-LON		Disponible	

Contact | Plan du site | Mentions légales | Infos éditeur et hébergeur