

# Le Langage XML

**Benedicte.Le-Grand@lip6.fr**

XML - 08/02/2002

1

## Points essentiels

- I. Présentation de XML
- II. Anatomie d'un document XML / Visualisation
- III. Feuilles de style
- IV. Vérification d'un document XML
  - document bien-formé
  - validation : DTD / Schéma
- V. Liens XML
- VI. Traitement de XML
- VII. XML Schemas
- VIII. Conclusion et références

XML - 08/02/2002

2

**www.Mcours.com**

Site N°1 des Cours et Exercices

Email: mymcours@gmail.com

## I. Présentation de XML

- I.1. Origine de XML
- I.2. Concepts de base
- I.3. Applications XML

XML - 08/02/2002

3

## I.1. Origine de XML

XML - 08/02/2002

4

**www.Mcours.com**

Site N°1 des Cours et Exercices

Email: mymcours@gmail.com

## Qu'est-ce qu'XML ? Pourquoi s'y intéresser ?

- Importance de XML dans un Internet en évolution
- Faiblesses d'HTML
  - ➔ Inadapté au commerce électronique
- SGML (Standard Generalized Markup Language), et relation avec XML
- Améliorations apportées par XML par rapport à SGML et HTML
- Avantages de XML pour les non-programmeurs

XML - 08/02/2002

5

## Évolution du World Wide Web

- Des simples pages Web ...
- ... au commerce électronique
  - Pas seulement commander à partir d'une page Web ([www.amazon.com](http://www.amazon.com))
  - Utiliser l'Internet pour regrouper les différentes parties d'une entreprise en une seule entité
  - Échange d'information sur des transactions financières (transactions de cartes de crédit, bancaires, etc.)
  - Échange de données médicales (entre médecins, hôpitaux, etc.)
  - Distribution de logiciels par le Web (pas d'installation, modularisation des logiciels et paiement des parties qu'on utilise)

XML - 08/02/2002

6

## Limites de HTML

- HTML a été et reste un succès fantastique
- Conçu pour afficher du texte dans un browser Web.
- Reproches usuels (justifiés ou non !)
  - Manque de vérification syntaxique (pas de validation)
  - Manque de structure
  - HTML non-orienté contenu : pas d'info sémantique
  - Pas international
  - Inadapté à l'échange de données
  - Absence d'un mécanisme de liens robuste
  - Non réutilisable
  - Non extensible

XML - 08/02/2002

7

## Limites de TeX, PDF, RTF

- TeX : communautés scientifiques
- PDF : Adobe
- RTF : Microsoft
  
- Formats propriétaires
- Non ouverts
- Non standardisés

XML - 08/02/2002

8

## SGML (Standard Generalized Markup Language)

- Objectif : stockage de données indépendant de tout vendeur ou package logiciel
- Meta language : description de langages de markup
  - HTML est une application de SGML
  - Très nombreux langages de markup définis à partir de SGML (QML : FAQ markup language)
- Déclaration SGML : distinction données / markup

XML - 08/02/2002

9

## Pourquoi pas SGML ?

- Tentative de « SGML sur le Web »
- Échec
  - SGML trop compliqué pour être utilisé à grande échelle sur le Web
  - Investissement trop important en temps, outils et formation

XML - 08/02/2002

10

## Pourquoi XML ?

- XML utilise les fonctionnalités de SGML dont il a besoin et incorpore les leçons apprises d'HTML
- XML peut-être utilisé avec les protocoles (HTTP, MIME) et les mécanismes (URLs) du Web
- Supporte une grande variété d'applications
- Compatible avec SGML
- Facile d'écrire des programmes qui traitent des docs XML (Java, SmallTalk, C, C++, JavaScript, Tcl, Perl, Python ...)
- Documents XML faciles à créer
- Lisibilité des documents XML

XML - 08/02/2002

11

## Qu'ajoute XML à SGML/HTML ?

- Modularité : combinaisons de fragments de DTDs et de documents
- Extensibilité : cible du lien pas nécessairement dans l'objet (pas forcément de droits d'écriture), nouvelles applications
- Distribution : notion de document composite, dont le contenu est créé automatiquement : séparation contenu / présentation
- Internationalité : basé sur Unicode (et non ASCII)
- Orienté données : doit être facilement lisible par les machines (création facile par les machines, pour le commerce électronique)
- But : faciliter le traitement automatisé des données

XML - 08/02/2002

12

## XML est-il réservé aux programmeurs ?

- Nouvelles applications utilisables avec des données XML
  - ➔ Les données doivent être disponibles en XML pour en bénéficier !
- XML peut être encore plus simple qu'HTML !
  - Pas de balises prédéfinies
  - Choix du nombre, des noms des balises

## I.2. Concepts de base

## Qu'est-ce qu'XML ?

- Extensible Markup Language
- Une syntaxe pour les documents
- Un « meta-markup language »
- Un langage structurel et sémantique, pas un langage de formatage
- Pas seulement pour les pages Web

## Meta Markup Language

- Pas comme HTML, LaTeX : création de nouvelles applications
- Création des balises en fonction des besoins
- Les balises créées peuvent être documentées dans une Document Type Definition (DTD)
- Syntaxe meta pour des langages de markup spécifiques comme MusicML, MathML, CML

## Description de la structure et de la sémantique, pas du formatage

- Les documents XML forment un arbre
- Éléments et attributs reflètent le contenu d'un élément
- Le formatage peut être ajouté avec une feuille de style

## Description d'une chanson en HTML

```
<dt>Hot Cop
<dd>by Jacques Morali, Henri Belolo, and Victor Willis
<ul>
<li>Producer: Jacques Morali
<li>Publisher: Polygram Records
<li>Length: 6:20
<li>Written: 1978
<li>Artist: Village People
</ul>
```

## Description d'une chanson en XML

```
<SONG>
  <TITLE>Hot Cop</TITLE>
  <COMPOSER>Jacques Morali</COMPOSER>
  <COMPOSER>Henri Belolo</COMPOSER>
  <COMPOSER>Victor Willis</COMPOSER>
  <PRODUCER>Jacques Morali</PRODUCER>
  <PUBLISHER>Polygram Records</PUBLISHER>
  <LENGTH>6:20</LENGTH>
  <YEAR>1978</YEAR>
  <ARTIST>Village People</ARTIST>
</SONG>
```

## Pour quoi utilise-t'on XML ?

- Langages de markup spécifiques à un domaine
- Données auto-descriptives
- Échange de données entre applications
- Structuration et intégration de données

## Langages de markup spécifiques à un domaine

- Pas de format propriétaire
- Création de nouveaux langages pour des domaines spécifiques

## Données auto-descriptives

- Beaucoup de données sont perdues à cause de problèmes de format
- XML est très simple
- XML est auto-descriptif
- XML est bien documenté

## Échange de données entre applications

- Problème des applications distribuées : pas de standard uniforme pour transférer des données structurées
- Standards utilisés par les applications souvent incompatibles
- Exemple : commerce électronique

## Structuration et intégration de données

- Spécification de relations entre les éléments
- Assemblage de données à partir de sources multiples

## I.3. Applications XML

## Applications XML

- Application XML : langage de markup spécifique utilisant la meta-syntaxe XML
- Les différentes applications XML ont leurs propres syntaxes et vocabulaires, avec des contraintes spécifiques

## Applications XML "réelles"

- Scalable Vector Graphics (SVG)
- The Open Software Description (OSD)
- The Internet Open Trading Protocol (OTP)
- The Java Speech Markup Language (JSML)
- The Health Level 7 (HL7)
- The Open Financial Exchange Specification (OFX)
- The Handheld Device Markup Language (HDML)
- The Channel Definition Format (CDF)
- Visual XML (VXML)
- The Signed Document Markup Language (SDML)
- The Translation Memory Exchange (TMX)

## Applications XML "réelles" (2)

- The Platform for Privacy Preferences (P3P)
- WebBroker
- The Development Markup Language (DML)
- The XML Query Language (XML-QL)
- The Bioinformatic Sequence Markup Language (BSML)
- The Chemical Markup Language (CML)
- The Synchronized Multimedia Integration Language (SMIL)
- The Mathematics Markup Language (MathML)
- ...

## Points essentiels

- I. Présentation de XML
- II. Anatomie d'un document XML / Visualisation
- III. Feuilles de style
- IV. Vérification d'un document XML
  - document bien-formé
  - validation : DTD / Schéma
- V. Liens XML
- VI. Traitement de XML
- VII. XML Schemas
- VIII. Conclusion et références

## II. Anatomie d'un document XML

- II.1. Étude d'un document XML
- II.2. Visualisation de XML dans des browsers

## Anatomie d'un document XML

- Markup
- Composants d'un document XML
- Extension standard : .xml
- Édition possible avec tout éditeur de texte

XML - 08/02/2002

31

## Markup

- Ajout de caractères à un morceau d'information, qui peut être utilisé pour traiter cette information.
  - Très simple : séparateurs de champs pour importer dans une BD (virgules)
  - Très riche : Text Encoding Initiative (versions, traductions, interprétation de documents historiques)
- Orienté contenu / orienté présentation
- Caractères de markup en XML : < & > ` ` "
- Tout ce qui n'est pas du markup est du contenu

XML - 08/02/2002

32

## II.1. Étude d'un document XML

XML - 08/02/2002

33

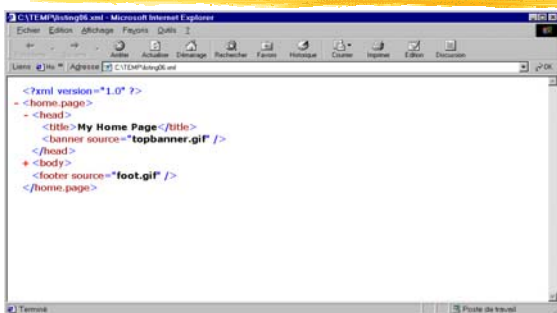
## Exemple de document XML

```
<?xml version="1.0"?>
<home.page>
  <head>
    <title>
      My Home Page
    </title>
    <banner source="topbanner.gif"/>
  </head>
  <body>
    <main.title>
      Welcome to My Home Page
    </main.title>
    <rule/>
    <text>
      <para>
        Sorry, this home page is still
        under construction. Please come
        back soon!
      </para>
    </text>
  </body>
  <footer source="foot.gif"/>
</home.page>
```

XML - 08/02/2002

34

## Affichage avec Internet Explorer 5



XML - 08/02/2002

35

## Déclaration XML (ligne 1)

```
<?xml version="1.0"?>
```

- Identification de la suite comme du code XML
- Trois attributs possibles :
  - version : numéro de version du standard XML auquel le document se conforme (1.0). (Obligatoire)
  - encoding : nom de l'encodage de caractères employé dans le fichier (UTF-8, UTF-16, ...)
  - standalone : vaut "no" si le document XML fait référence à une DTD ou à des entités définies dans un autre fichier, vaut "yes" sinon.
- Déclaration XML = instruction de traitement (Processing Instruction)

XML - 08/02/2002

36

## Élément racine (lignes 2 à 23)

- Premier élément déclaré dans le document XML
- 1 seul élément racine
- Tous les autres y sont inclus
- Les éléments doivent être parfaitement imbriqués
  
- Un élément XML est constitué de 3 parties
  - Une balise de début
  - Du contenu (texte ou éléments)
  - Une balise de fin (nom identique à la balise de début)

XML - 08/02/2002

37

## Exemple d 'élément

```
<title>My Home Page</title>
```

- Balise de début : <title>
- Contenu : « My Home Page »
- Balise de fin : </title>

XML - 08/02/2002

38

## Élément vide (ligne 13)

- Il faut spécifier clairement qu'un élément est vide :

```
<element_vide/>
```

- Autre possibilité :

```
<element_vide></element_vide>
```

- Un élément vide contient des informations dans ses attributs

XML - 08/02/2002

39

## Attributs (lignes 7 et 22)

- Nécessairement contenus dans la balise de début
- Complément d'information sur l'élément
- Syntaxe : nom=valeur  
    <nom\_element nom\_attribut="valeur\_attribut">  
        Ou  
        <nom\_element nom\_attribut='valeur\_attribut'>
- Guillemets/apostrophes obligatoires pour la valeur
- Fusion des ensembles d'attributs pour un élément donné
- La valeur d'un attribut ne peut pas être structurée
- Un élément ne peut pas avoir deux attributs de même nom

XML - 08/02/2002

40

## Règles de nommage

- Au moins une lettre : a à z ou A à Z
- Peut commencer par \_ ou :
- La 1ère lettre peut être suivie par une ou plusieurs lettres, chiffres, tirets, underscores, points, et l'ensemble des caractères Unicode.  
    ➔ <http://www.w3.org/XML/REC-xml>
- Pas d'espaces ni de tabulations
- Seuls signes de ponctuation autorisés : tiret et point

XML - 08/02/2002

41

## Commentaires

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

- Tout le contenu sera ignoré par le processeur XML
  - Pas d'espace dans les balises <!-- et -->
  - Pas de commentaire à l'intérieur d'autre markup
- ```
<para <!-- aïe !--> > bonjour </para>
```

XML - 08/02/2002

42

## 5 entités prédéfinies (pour les caractères de markup)

- & → &amp; ou &#38;#38;
- ' → &apos; ou &#39;
- > → &gt; ou &#62;
- < → &lt; ou &#38;#60;
- " → &quot; ou &#34

XML - 08/02/2002

43

## Sections CDATA

- Pour garder le texte tel qu'il est :  
`<![CDATA[ texte <5 lignes> non regardé &" par le processeur]]>`
- Rien n'est alors reconnu comme du markup
- Exemple : pour inclure du code applicatif (JavaScript, Perl, etc.) dans le code XML (autre manière : Processing Instruction)

XML - 08/02/2002

44

## Processing Instructions

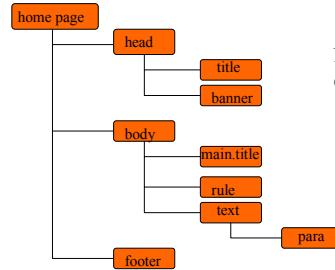
- Déclaration XML  
`<?xml version="1.0"?>`
- Instructions pour des applications spécifiques  
`<?xml version="1.0"?>`  
`<home.page>`  
`<para>This is text containing two processing instructions,`  
`<?javascript I can put whatever I like here?>`  
`<?perl And I can put whatever I like here too?>`  
`one for each interpreter.</para>`  
`</home.page>`

XML - 08/02/2002

45

## Structure logique du document

- Structure arborescente



Notions de parent, enfant et "sibling"

XML - 08/02/2002

46

## II.2. Visualisation de XML dans des browsers

- Netscape 4
- Netscape Mozilla 5
  - [www.mozilla.org](http://www.mozilla.org)
- Citic Information Technologies Doczilla
  - <http://www.doczilla.com>
- Internet Explorer 5

XML - 08/02/2002

47

## Internet Explorer 5 et XML

- Entièrement conforme à la spécification XML 1.0
- Data Islands
- Visualisation directe avec XSL et CSS
- Support des espaces de nommage
- Meilleures performances

XML - 08/02/2002

48



## IE 5 Data Islands : XML inclus dans HTML

```
<HTML>
<HEAD><TITLE>XML Object Model in Explorer
5.0</TITLE>
</HEAD>
<BODY>
<H2>My favorite musicians</H2>
<HR>
<XML id="myfavs">
<musicians>
<musician>
<name>Joey Baron
</name>
<instrument>drums
</instrument>
<NrOfRecordings>1
</NrOfRecordings>
</musician>
<musician>
<name>Bill Frisell
</name>
<instrument>guitar
</instrument>
</musician>
</musicians>
</XML>
</BODY>
</HTML>
```

XML - 08/02/2002

49

## Visualisation en utilisant le DOM

- Modèle accessible par JavaScript, VBScript, C++, Java

```
function output_doc(node)
{
  if (node.nodeType == 0)
  {
    if (node.nodeName == "musicians")
    {
      document.write("<TABLE BORDER='1' CELLSPACING='5'>");
      traverse(node);
      document.write("</TABLE>");
    }
    else if (node.nodeName == "musician")
    {
      document.write("<TR>");
      traverse(node);
      document.write("</TR>");
    }
  }
  ...
}
```

XML - 08/02/2002

50

## Visualiser XML directement

- Possibilité d'ouvrir tout fichier XML dans IE5
- Feuille de style XSL par défaut

XML - 08/02/2002

51

## Exemple de visualisation directe

```
<?xml version="1.0" ?>
<musicians>
<musician>
<name>Joey Baron
</name>
<instrument>drums
</instrument>
<NrOfRecordings>1
</NrOfRecordings>
</musician>
<musician>
<name>Bill Frisell
</name>
<instrument>guitar
</instrument>
<NrOfRecordings>3
</NrOfRecordings>
</musician>
</musicians>
```



XML - 08/02/2002

52

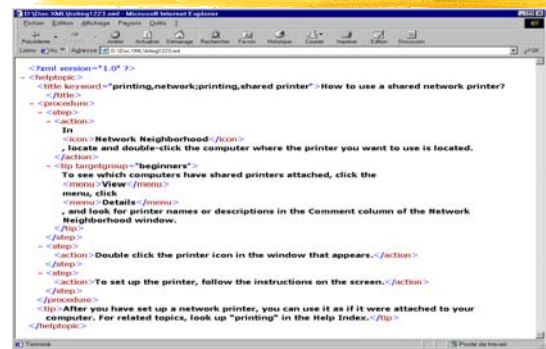
## Autre exemple

```
<?xml version="1.0" ?>
<helptopic>
<title keyword="printing,network,printing,shared printer">
How to use a shared network printer</title>
<procedure>
<step><action>In <icon>Network Neighborhood</icon>,
locate and double-click the computer where the printer
you want to use is located. </action>
<tip targetgroup="beginners">To see which computers have
shared printers attached, click the <menu>View</menu> menu,
click <menu>Details</menu>, and look for printer names or
descriptions in the Comment column of the Network Neighborhood window.</tip>
</step>
<step>
<action>Double click the printer icon in the window that appears.</action>
</step>
<step>
<action>To set up the printer, follow the instructions on the screen.
</action></step>
</procedure>
<tip>
After you have set up a network printer, you can use it as if
it were attached to your computer. For related topics,
look up &quot;printing&quot; in the Help Index.
</tip>
</helptopic>
```

XML - 08/02/2002

53

## Visualisation avec la feuille de style par défaut



## Points essentiels

- I. Présentation de XML
- II. Anatomie d'un document XML / Visualisation
- III. Feuilles de style
- IV. Vérification d'un document XML
  - document bien-formé
  - validation : DTD / Schéma
- V. Liens XML
- VI. Traitement de XML
- VII. XML Schemas
- VIII. Conclusion et références

XML - 08/02/2002

55

## III. Feuilles de style

XML - 08/02/2002

56

## Feuilles de style

Séparation du contenu et de la présentation

- **Interopérabilité** : affichage sur le Web, sur papier, BD, ...
- **Réutilisabilité/ durabilité** : le document ne devient pas obsolète avec l'évolution des techniques informatiques ; il pourra être traité par des applications inexistantes au départ

XML - 08/02/2002

57

## Feuilles de style

- Séparées du document XML
- CSS : Cascading Style Sheets
  - CSS1 : Level 1
  - CSS2 : Level 2
- XSL : Extensible Stylesheet Language
- DSSL : Document Style and Semantics Language

XML - 08/02/2002

58

## CSS ou XSL ?

- CSS : plus large support
- CSS : plus stable
- XSL : plus puissant

XML - 08/02/2002

59

## III.1. CSS

XML - 08/02/2002

60

## Feuille de style CSS

```
helptopic { display: block;
margin-top:3cm;
margin-left:2cm;
margin-right:2cm;
margin-bottom:6cm;
font-family:Verdana, Arial;
font-size:11pt;
padding:20pt; }

title {display: block;
font-size:20pt;
color:blue;
font-weight:bold;
text-align:center;
margin-bottom:30pt;
text-decoration:underline;}

procedure {display:block;
margin-bottom:30pt;}

step {display:block;
margin-bottom:18pt;}

action {display:block;
font-weight:bold;}

tip {display:block;
font-size:10pt;
margin-left:+1cm;
margin-top:12pt;
color:blue;}

icon {display:inline;
font-size:12pt;}

todo {display:inline;
color:red;}

menu {display:inline;
font-style:italic;}
```

XML - 08/02/2002

61

## Fichier XML correspondant

Après la déclaration XML, avant l'élément racine

```
<?xml version="1.0" ?>
<?xml:stylesheet type="text/css" href="helptopic.css" ?>
<helptopic>
<title keyword="printing,network;printing,shared printer">
...

```

XML - 08/02/2002

62

## Affichage avec la feuille de style CSS



## Propriétés de CSS

- Tout élément n'a pas besoin d'une règle
- Élément racine : au moins display: block

XML - 08/02/2002

64

## III.2. XSL

## Feuille de style XSL

```
<?xml version="1.0" ?>
<xml:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl" ?>

<!-- default behaviour -->
<xsl:template>
<xsl:apply-templates/>
</xsl:template>

<!-- specific behaviour -->
<xsl:template match="/">
<html>
<head>
<title>Using an XSL stylesheet </title>
</head>
<body bgcolor="#FFFFFF">
<xsl:apply-templates/>
</body>
</html>
</xsl:template>

<xsl:template match="title">
<h2>
<xsl:apply-templates/>
</h2>
</xsl:template>

<xsl:template match="procedure">
<OL>
<xsl:apply-templates/>
</OL>
</xsl:template>

<xsl:template match="step">
<LI>
<xsl:apply-templates/>
</LI>
</xsl:template>

<xsl:template match="action">
<B>
<xsl:apply-templates/>
</B><BR/>
</xsl:template>

<xsl:template match="helptopic/tip">
<H3>Tip!</H3>
<xsl:apply-templates/>
</xsl:template>

</xsl:stylesheet>
```

XML - 08/02/2002

65

XML - 08/02/2002

66

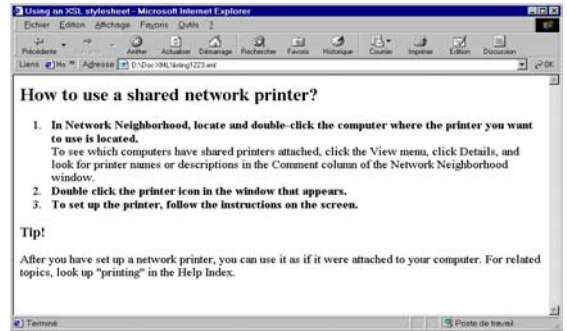
## Fichier XML correspondant

```
<?xml version="1.0" ?>
<?xml:stylesheet type="text/xsl" href="helptopic.xsl" ?>
<helptopic>
<title keyword="printing,network;printing,shared printer">
...
```

XML - 08/02/2002

67

## Affichage avec la feuille de style XSL



## Deux parties de XSL

- XSL-T : langage de transformation normalisé, pouvant transformer une DTD ou un arbre XML orienté "contenu" en une autre DTD ou arbre orienté "restitution", constitué d'objets formateurs
- XSL-FO : Objets formateurs

XML - 08/02/2002

69

## Templates

```
<HTML>
  <HEAD>
    <TITLE>
      XSL instructions to get the title
    </TITLE>
  </HEAD>
  <H1>XSL instructions to get the title</H1>
  <BODY>
    XSL instructions to get the data
  </BODY>
</HTML>
```

XML - 08/02/2002

70

## Instructions XSL

- Une feuille de style XSL est un document XML bien-formé
- Les instructions XSL sont des éléments XML particuliers
  - xsl:apply-templates
  - xsl:template
  - xsl:for-each
  - xsl:value-of
  - quelques autres

XML - 08/02/2002

71

## Points essentiels

- I. Présentation de XML
- II. Anatomie d'un document XML / Visualisation
- III. Feuilles de style
- IV. Vérification d'un document XML
  - document bien-formé
  - validation : DTD / Schéma
- V. Liens XML
- VI. Traitement de XML
- VII. XML Schemas
- VIII. Conclusion et références

XML - 08/02/2002

72

## IV. Vérification d'un document XML

- IV.1. Document bien-formé
- IV.2. Document valide

## IV.1. Document XML bien formé

- Contient 1 ou plusieurs éléments
- Un seul élément contient tous les autres (racine)
- Ses éléments sont correctement imbriqués les uns dans les autres
- Les noms des balises de début et de fin correspondent exactement
- Les noms des attributs n'apparaissent qu'une fois pour chaque élément
- Les valeurs des attributs sont entre guillemets ou apostrophes
- Les valeurs des attributs ne référencent pas d'entités externes
- Les entités sont déclarées avant d'être utilisées

## Ouvrir et fermer toutes les balises

- Différent de HTML
- Balises de début et de fin strictement identiques (casse comprise)

## Fermeture des balises vides

- `<BR/>`, `<HR/>` et `<IMG/>` au lieu de `<BR>`, `<HR>` et `<IMG>`

On peut utiliser `<BR></BR>`, `<HR></HR>` et `<IMG></IMG>` à la place

## Unique élément racine

- Un élément contient tous les autres éléments du document
- La déclaration XML n'est pas un élément

## Les éléments doivent être correctement imbriqués

- Si un élément contient la balise de début d'un autre élément, il doit aussi contenir sa balise de fin
- Les éléments vides peuvent apparaître n'importe où
- Tout élément non racine a un élément père

## Les valeurs d'attributs sont entre guillemets / apostrophes

- Correct :  
`<A HREF="http://metalab.unc.edu/xml/" >>`
- Incorrect :  
`<A HREF=http://metalab.unc.edu/XML>`

XML - 08/02/2002

79

## < et & utilisés seulement pour les balises de début et les entités

- Correct :  
`<H1>O'Reilly & Associates</H1>`
- Incorrect :  
`<H1>O'Reilly & Associates</H1>`
- Correct :  
`<CODE>for (int i =0; i &lt;= args.length; i++) </CODE>`
- Incorrect :  
`<CODE>for (int i =0; i <= args.length; i++) </CODE>`

XML - 08/02/2002

80

## Vérification de la "bonne-formation"

- Comment vérifier qu'un document XML est bien formé ?
- Analyseurs ("parsers") non-validants :
  - ➔ Logiciels qui vérifient la syntaxe des fichiers en fonctions des règles de "bonne-formation" vues précédemment

XML - 08/02/2002

81

## Analyseurs XML disponibles

- Robin Cover : site de référence sur SGML et XML  
<http://www.oasis-open.org/cover/>
- James Tauber  
<http://www.xmlsoftware.com/parsers/>

XML - 08/02/2002

82

## Services de validation online

- RUWF : vérificateur de syntaxe XML  
<http://www.xml.com/xml/pub/tools/ruwf/check.html>
- Richard Tobin  
<http://www.cogsci.ed.ac.uk/~richard/xml-check.html>

XML - 08/02/2002

83

## Exemple

```
<?xml version="1.0"?>
<?protest objid="15678" ?>
<helptopic>
<title keyword="printing,network;printing,shared
printer">How
to use a shared network printer?</title>
<procedure>
<step><action>In <icon>Network Neighborhood</icon>,
locate and
double-click the computer where the printer you want to
use is
located. </action>
<tip targetgroup="beginners">To see which computers
have
shared printers attached, click the <menu>View</menu>
menu,
click <menu>Details</menu>, & look for printer names or
descriptions in the Comment column of the Network
Neighborhood window.</tip>
</step>
<step>
<action>&doubleclick; the printer icon in the window that
appears.</action>
</step>
<step>
<action>
To set up the printer, <xref linkend="td45">follow the
instructions</xref> on the screen.
</step>
</procedure>
<rule form="double">
<tip>
<P>After you have set up a network printer, you can use it
as
if it were attached to your computer. For related topics, look
up
&quot;printing&quot; in the Help Index.
</p>
</tip>
</helptopic>
</step>
```

XML - 08/02/2002

84

## Document bien-formé

```
<?xml version="1.0" ?>
<?protext objid="15678" ?>
<!DOCTYPE helptopic [
<!ENTITY doubleclick "Double-click">
]>
<helptopic>
<title keyword="printing,network;printing,shared
printer">How
to use a shared network printer?</title>
<procedure>
<step><action>In <icon>Network
Neighborhood</icon>, locate and
double-click the computer where the printer you want
to use is
located. </action>
<tip targetgroup="beginners">To see which computers
have
shared printers attached, click the
<menu>View</menu> menu,
click <menu>Details</menu>, &amp; look for printer
names or
descriptions in the Comment column of the Network
Neighborhood window.</tip>
</step>
```

XML - 08/02/2002

85

## IV.2. Document valide

- Document Type Definition
- Sous-ensemble interne
- Sous-ensemble externe
- Développement d'une DTD

XML - 08/02/2002

86

## Intérêt de la validation

- Facilite l'échange de fichiers, car ils utilisent la même grammaire
- Validité pas nécessaire pour les browsers

XML - 08/02/2002

87

## Importance de la validation

- Assure que les données sont correctes avant de les donner à un programme
- Assure qu'un format donné est suivi
- Établit ce qui doit être supporté
- Tous les documents n'ont pas besoin d'être valides ; être bien-formé peut suffire

XML - 08/02/2002

88

## DTD et validation

- DTD : modèle de la structure du contenu d'un document XML
  - ce qui est permis ou non dans un élément
- HTML a une seule DTD
- XML permet la création de nos propres DTDs, en fonction des applications
- DTD stricte ou non
- L'auteur de la DTD choisit ce qui est autorisé ou non

XML - 08/02/2002

89

## DTD et validité

- Liste des éléments, attributs et entités contenus dans le document XML, ainsi que les relations entre eux
- Validation : Vérification de la conformité d'un document XML à une DTD donnée
- Bonne-formation = prérequis pour la validité

XML - 08/02/2002

90

## DTD interne

- Toutes les déclarations sont internes
- Pas de traitement de documents externes
- Déclaration XML :

```
<?xml version="1.0" standalone="yes"?>
```

- Les déclarations d'entités externes doivent être contenues dans la DTD interne

XML - 08/02/2002

91

## Exemple de document XML standalone

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE CATALOG [
<ELEMENT CATALOG (PRODUCT+)>
<ELEMENT PRODUCT (SPECIFICATIONS+, PRICE+, NOTES?)>
<!ATTLIST PRODUCT NAME CDATA #REQUIRED>
<ELEMENT SPECIFICATIONS (#PCDATA)>
<!ATTLIST SPECIFICATIONS SIZE CDATA #REQUIRED
COLOR CDATA #REQUIRED>
<ELEMENT PRICE (#PCDATA)>
<!ATTLIST PRICE WHOLESAL NMTOKEN #REQUIRED
RETAIL NMTOKEN #REQUIRED
SALES.TAX NMTOKEN #IMPLIED>
<ELEMENT NOTES (#PCDATA)>
]>
<CATALOG>
<PRODUCT NAME="T-shirt">
<SPECIFICATION SIZE="XL" COLOR="WHITE"/>
<PRICE WHOLESAL="9.95" RETAIL="19.95" SALES.TAX="2.56"/>
<NOTES>Dibert</NOTES>
</PRODUCT>
<PRODUCT NAME="Shirt">
<SPECIFICATION SIZE="38" COLOR="BLACK"/>
<PRICE WHOLESAL="69.95" RETAIL="79.95" SALES.TAX="4.54">Euro</PRICE>
</PRODUCT>
</CATALOG>
```

XML - 08/02/2002

92

## Exemple de DTD externe

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE CATALOG SYSTEM "catalog.dtd">
<CATALOG>
<PRODUCT NAME="T-shirt">
<SPECIFICATION SIZE="XL"
COLOR="WHITE"/>
<PRICE WHOLESAL="9.95"
RETAIL="19.95" SALES.TAX="2.56"/>
<NOTES>Dibert</NOTES>
</PRODUCT>
<PRODUCT NAME="Shirt">
<SPECIFICATION SIZE="38"
COLOR="BLACK"/>
<PRICE WHOLESAL="69.95"
RETAIL="79.95"
SALES.TAX="4.54">Euro</PRICE>
</PRODUCT>
</CATALOG>
```

```
<!-- catalog.dtd -->
<ELEMENT CATALOG (PRODUCT+)>
<ELEMENT PRODUCT (SPECIFICATIONS+,
PRICE+, NOTES?)>
<!ATTLIST PRODUCT NAME CDATA
#REQUIRED>
<ELEMENT SPECIFICATIONS (#PCDATA)>
<!ATTLIST SPECIFICATIONS SIZE CDATA
#REQUIRED
COLOR CDATA #REQUIRED>
<ELEMENT PRICE (#PCDATA)>
<!ATTLIST PRICE WHOLESAL NMTOKEN
#REQUIRED
RETAIL NMTOKEN #REQUIRED
SALES.TAX NMTOKEN #IMPLIED>
<ELEMENT NOTES (#PCDATA)>
```

XML - 08/02/2002

93

## a. Déclaration d'entités

- Contenues dans la DTD qui accompagne le document XML
- Entité : unité de stockage
- Un document XML peut être constitué de différents fichiers ou objets dont chacun est une entité

XML - 08/02/2002

94

**www.Mcours.com**

Site N°1 des Cours et Exercices

Email: mymcours@gmail.com

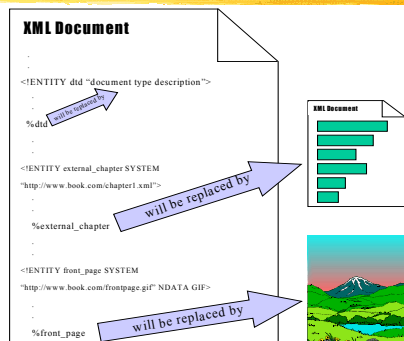
## Déclaration d'entités

- Entité interne
  - Entité analysable : contenu = texte de remplacement
    - Entité générale (document)
    - Entité paramètre (DTD)
- Entité externe
  - Entité analysable : contenu = texte de remplacement
    - Entité générale
    - Entité paramètre (DTD)
  - Entité non-analysable : associée à une notation

XML - 08/02/2002

95

## Exemple



96



## Entités générales internes

- Symboles pouvant être définis dans une DTD et utilisés dans un document XML comme raccourcis d'écriture.
- Déclaration d'entité interne :  
`<!ENTITY nom "texte de remplacement">`
- Remplacement de `&nom;` par le texte de remplacement lors du traitement du code XML

XML - 08/02/2002

97

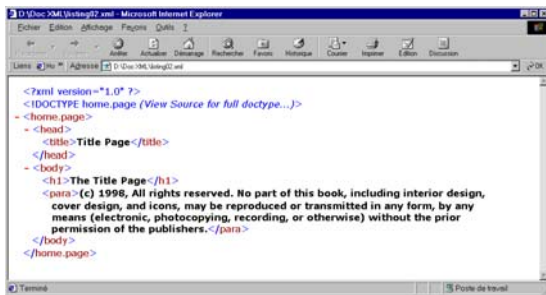
## Exemple d'entité interne

```
<?xml version="1.0"?>
<!DOCTYPE home.page [
  <!ENTITY rights "All rights reserved. No part of this book, including interior
  design, cover design, and icons, may be reproduced or transmitted in any
  form,
  by any means (electronic, photocopying, recording, or otherwise) without the
  prior permission of the publishers.">
]>
<home.page>
  <head>
    <title>Title Page</title>
  </head>
  <body>
    <h1>The Title Page</h1>
    <para>(c) 1998, &rights;</para>
  </body>
</home.page>
```

XML - 08/02/2002

98

## Affichage avec Internet Explorer 5



XML - 08/02/2002

99

## Entités paramètres

- Raccourci pour les déclarations de markup
- Syntaxe différente des entités caractères  
➔ Signe % (au lieu de &)
- Exemple :  
`<!ENTITY % titre "H1 | H2 | H3 | H4 | H5 | H6 ">`  
`<!ENTITY % contenu "(%titre | %texte)*">`  
`<!ELEMENT BODY %contenu>`

XML - 08/02/2002

100

## Déclaration des éléments et attributs dans la DTD interne

- Déclaration d'entités internes (sortes de macros)
- Déclaration d'éléments et d'attributs dans la DTD interne
- Premiers pas dans le monde de la modélisation d'information
- Contrôle et vérification du contenu de documents XML conformes à des DTDs

XML - 08/02/2002

101

## b. Déclaration d'éléments

- `<!ELEMENT nom contenu>`
- nom : nom standard XML (cf. règles de nommage)
- contenu :
  - Mot-clé EMPTY
  - Mot-clé ANY
  - Modèle de contenu ("content model") : séquence et répétition des éléments contenus dans cet élément

XML - 08/02/2002

102

## Exemple de message email

```
<?xml version="1.0"?>
<message>
  <header>
    <date>14 May 1998</date>
    <From>Me</From>
    <To>You</To>
    <Subject>Test Message</Subject>
  </header>
  <Body> ...
</Body>
  <Sig>Some smart saying
</Sig>
</message>
```

XML - 08/02/2002

103

## DTD pour ce message email

```
<!ELEMENT message (header, body, sig) >
<!ELEMENT header (date, from, to, subject) >
<!ELEMENT body (#PCDATA) >
<!ELEMENT sig (#PCDATA) >
<!ELEMENT date (#PCDATA) >
<!ELEMENT from (#PCDATA) >
<!ELEMENT to (#PCDATA) >
<!ELEMENT subject (#PCDATA) >
```

XML - 08/02/2002

104

## Documents valides / invalides

- DTD :  
`<!ELEMENT GREETING #(PCDATA)>`
- Valide :  
`<GREETING>texte sans markup</GREETING>`
- Invalide :  
`<GREETING><sometag>...</sometag></GREETING>`
- Invalide :  
`<GREETING>  
 <GREETING>texte</GREETING>  
</GREETING>`

XML - 08/02/2002

105

## Déclaration d'éléments vides

- Élément vide (rappel) :  
`<element_vide/>`  
Ou  
`<element_vide></element_vide>`
- Déclaration :  
`<!ELEMENT element_vide EMPTY>`

XML - 08/02/2002

106

## Déclaration d'éléments non restreints

- Peut contenir tout type d'élément déclaré dans la DTD
- Déclaration :  
`<!ELEMENT element_quelconque ANY>`

XML - 08/02/2002

107

## Déclaration de modèles de contenu

- Spécification des éléments autorisés dans d'autres éléments, de leur ordre et de leur fréquence
- Séquences
- Choix
- Combinaisons
- Occurrences
- Contenu caractère

XML - 08/02/2002

108

## Séquences d'éléments

- Liste des éléments possibles, contenus dans des parenthèses et séparés par des virgules

<!ELEMENT repas (entrée, plat, dessert)>

- Un repas doit être composé d'une entrée, suivie d'un plat puis d'un dessert.
- Les trois éléments doivent être présents dans l'élément repas

## Choix d'éléments

- Choix indiqué par une ligne verticale |

<!ELEMENT choix (celui-ci | celui-là)>

- Un élément choix consiste en un élément celui-ci ou un élément celui-là
- Seul un élément peut être sélectionné, quelle que soit la longueur de la liste

## Combinaison séquences / choix

- Regroupements pour combiner séquences et choix :

<!ELEMENT combi (entrée1 | entrée 2), (plat1 | plat2)>

- Élément combi :

entrée1 plat1 ou entrée1 plat2  
ou entrée2 plat1 ou entrée2 plat2

## Attention aux ambiguïtés !

<!ELEMENT ambig ((je, tu) | (je, vous))>

- Qu'est-ce qui vient après "je" ?
- Correction :  
<!ELEMENT non\_ambig (je, (tu | vous))>
- Ne pas hésiter à utiliser les parenthèses  
<!ELEMENT ambig (je, tu, il | elle)>  
<!ELEMENT non\_ambig (je, tu, (il | elle))>

## Indicateurs d'occurrences d'éléments

- pas indicateur d'occurrence :

→ exactement 1 fois

- ? → 0 ou 1 fois

- \* → 0 ou plus

- + → 1 ou plus

## Contenu caractère

- Mot-clé PCDATA lorsque du texte (et seulement du texte) est contenu dans un élément
- Précédé de #
- Un Parseable Character ne contient pas de markup

## Contenu mixte

- Utilisation particulière de la grammaire :
  - Un ensemble d'alternatives
  - #PCDATA en premier
  - Toujours \* à la fin (sauf quand uniquement #PCDATA)

<!ELEMENT mixte (#PCDATA | entrée | plat)\*>

## c. Déclaration d'attributs

- Syntaxe :  
<!ATTLIST nom\_element definition\_attr>
- Définition d'attribut :  
nom\_attrib type\_attrib  
ou  
nom\_attrib type\_attrib valeur\_par\_défaut

## Types d'attributs

- Chaîne de caractères (CDATA)  
<!ATTLIST livre auteur CDATA>
- Un ou plusieurs "tokens" significatifs pour XML
  - ID : symbole unique sur l'ensemble du document, commençant par une lettre
  - IDREF : valeur de l'attribut = valeur de l'attribut ID d'un autre élément
  - IDREFS : comme IDREF, mais il peut y avoir plusieurs valeurs
  - ENTITY, ENTITIES, NMTOKEN, NMTOKENS
- Attribut énuméré  
<!ATTLIST peinture couleur (rouge|vert|jaune)>

## Valeur par défaut d'un attribut

- #IMPLIED : l'application traitant le document fournira la valeur par défaut
- #REQUIRED : l'auteur du document doit définir l'attribut en question
- #FIXED : l'attribut ne peut prendre que la valeur mentionnée
- Une valeur compatible avec le type associé à l'attribut : sans indication explicite, l'attribut prendra cette valeur

## Vérification de la validité

- Vérification de la DTD
- Conformité entre le document et les règles définies dans la DTD

## Analyseur XML for Java

- Analyseur XML validant écrit en Java
- <http://alphaworks.ibm.com/tech/xml4j>
- Le package contient des classes et des méthodes pour analyser, générer, manipuler et valider des documents XML

## Outils de validation en ligne

- <http://www.stg.brown.edu/service/xmlvalid/>
- <http://www.cogsci.ed.ac.uk/~richard/xml-check.html>

XML - 08/02/2002

121

## Application

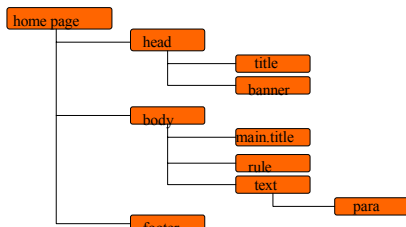
- Étude d'une DTD
  - Nombre d'éléments définis ?
  - Comment rendre l'*auteur* facultatif ?
- Document XML
  - Nombre de sous-éléments d'*article* ?
- Document bien-formé / valide ?

XML - 08/02/2002

122

## Développement d'une DTD

- À partir d'un document XML
- Hiérarchie des éléments



XML - 08/02/2002

123

## Développement d'une DTD

- "À la main"
- Identification des éléments
  - 3 types de markup
    - Contenu (ce que l'information représente)
    - Structure (groupements en listes, paragraphes, etc.)
    - Présentation (à éviter)
- Structuration des éléments
  - Arrangement hiérarchique
  - Choix de la racine, des enfants de la racine, ...
- Assignation d'attributs

XML - 08/02/2002

124

**www.Mcours.com**

Site N°1 des Cours et Exercices

Email: [mymcours@gmail.com](mailto:mymcours@gmail.com)

## Éléments / Attributs ?

- Éternelle question ...
- Historiquement :
  - Attributs pour les propriétés abstraites, intangibles (couleur, taille, etc.)
  - Éléments pour les composants physiques
- Éléments si l'information doit être structurée
- Plus facile d'éditer les éléments que les attributs dans des éditeurs XML
- Vérification plus facile de la valeur d'un attribut
- Attributs "fusionnables" entre différents documents XML
- Plus facile de fragmenter un document XML en fonction de la structure des éléments qu'en fonction de la valeur des attributs

XML - 08/02/2002

125

## Éléments / Attributs (2)

- Attributs pour les meta données  
Éléments pour les données
- Éléments si on veut lire les informations
- Attributs appropriés aux IDs, URLs, références, et toutes les infos non directement pertinentes pour l'utilisateur

XML - 08/02/2002

126

## Espaces de nommage

- Un document XML peut utiliser des balises définies dans différentes DTDs
- Pour utiliser des symboles définis ailleurs, le document XML doit "importer" l'espace de noms correspondant : lève les ambiguïtés éventuelles des intitulés de balise
- Ajout d'un attribut approprié  

```
<?xml version="1.0" encoding="ISO-8859-1 ?>  
<article xmlns:html=http://www.w3.org/Profiles/XHTML-transitional>  
  Balises préfixées  
<html:img html:src="..." />
```

XML - 08/02/2002

127

## Points essentiels

- I. Présentation de XML
- II. Anatomie d'un document XML / Visualisation
- III. Feuilles de style
- IV. Vérification d'un document XML
  - document bien-formé
  - validation : DTD / Schéma
- V. Liens XML
- VI. Traitement de XML
- VII. XML Schemas
- VIII. Conclusion et références

XML - 08/02/2002

128

## V. Liens XML

XML - 08/02/2002

129

## Langages de lien et d'adressage

En cours de spécification au sein de 3 documents :

- Xpath** : expression de chemins à l'intérieur d'un document XML
- XLL : Extensible Linking Language**
  - Xpointer** : langage d'adressage des contenus d'un document XML
  - XLink** : description des liens entre objets

XML - 08/02/2002

130

## Création de liens XML

- Différences entre les liens HTML / XML
- Variété de types d'hyperliens en XML
  - Liens entre plusieurs points
  - Liens vers des documents read-only
  - Description de liens
- Attributs des liens pour contrôler quand et comment les liens sont activés
- Ajout de comportement hyperlien

XML - 08/02/2002

131

## Hyperliens HTML

- Association entre :
  - deux parties de texte
  - du texte et un objet
  - un objet et du texte
- Lien HTML : lien entre une source et une cible
  - Cible : page HTML, ou un élément d'une page HTML, nommé avec l'attribut NAME

XML - 08/02/2002

132

## Exemple de lien HTML

Code HTML pour la source du lien :

```
<H3>lien simple</H3>
<P>Ce <A HREF=http://lien_simple.html#S1>lien</A> pointe vers une
section particulière.</P>
```

Code HTML dans le document cible :

```
<H4><A NAME="S1">section cible</A></H4>
<P>Cette section est celle vers laquelle je veux pointer.</P>
```

## Hyperliens XML

- XLink : XML Linking Language
- Sources → "linking elements"
  - car les liens sont bidirectionnels
  - Les linking elements contiennent des "locators"
- Cibles → ressources
  - données, résultats de requête dans une BD, etc.
- On ne "suit" plus un lien, on le "traverse"

## Liens simples

- Très proches des liens HTML
- 1 seul locator
- 1 seule direction

```
<lien.simple xlink:form="simple"
href=http://toto.xml>aller voir</lien.simple>
```

## Liens simples

attribut	valeurs possibles	signification
xlink:type	simple	
xlink:href	une URL	destination (unique) du lien le document destination apparaîtra dans une nouvelle fenêtre
	new	le document destination apparaîtra dans la même fenêtre que le document courant (comportement similaire aux liens HTML)
	replace	le document destination apparaîtra dans le document courant, à la place de l'élément constituant le lien (comportement similaire aux balises HTML <img> et <object>)
	embed	
xlink:show	undefined	l'application traitant le document choisira le traitement approprié

## Liens simples (2)

attribut	valeurs possibles	signification
xlink:actuate	onLoad	le document destination apparaîtra dès le chargement du document courant (comportement similaire aux balises HTML <img> et <object>)
	onRequest	le document destination apparaîtra quand l'utilisateur activera le lien (généralement en "cliquant" dessus)
	undefined	l'application traitant le document choisira le moment approprié
xlink:title	une chaîne de caractères	le texte fourni pourra servir d'indice supplémentaire, par exemple en étant affiché quand la souris se trouve sur l'élément contenant le lien

## Liens simples - exemple

```
<article xmlns:xlink="http://www.w3.org/1999/xlink" >
<titre>Un journaliste accuse, un policier d&#233;ment</titre>
<auteur xlink:href="AlainHome.xml" xlink:type="simple"
xlink:show="replace" xlink:actuate="OnRequest"
xlink:title="Homepage">Alain Connu</auteur>
...
<text>
<img>
<html:img src="photo.gif" xlink:href="Details.xml"
xlink:type="simple" xlink:show="replace"
xlink:actuate="OnRequest" xlink:title="details" />
</img>
</text>
</article>
```

## Référence croisée avec des liens simples

```
<?xml version="1.0"?>
<doc>
  <para id="idea1">
    <xref href="idea2">A first impression of the subject
    matter would inevitably lead to the conclusion that ...
    </xref>
  </para>
  <para id="idea2">
    <xref href="idea1">Contrary to our earlier conclusion,
    it would appear that the evidence points to a different ...
    </xref>
  </para>
</doc>
```

XML - 08/02/2002

139

## Liens étendus

- Liens pas nécessairement contenus dans l'un des fichiers XML reliés par le lien
- Relier ensemble plusieurs ressources (pas seulement une relation 1-1 comme en HTML)
- Relier des ressources qui ne contiennent pas les liens elles-mêmes (graphiques, sons, documents read-only, non modifiables pour inclure les liens)
- Filtrage dynamique, addition, modification de liens
- Une application peut traiter les liens selon ses propres besoins

XML - 08/02/2002

140

## Liens étendus

attribut	valeur	signification
<b>xlink:type</b> (obligatoire)	extended	définition de l'élément englobant formant un lien étendu
<b>xlink:role</b>	identificateur (avec préfixe si l'identificateur est défini dans un autre fichier)	l'identificateur utilisé ici peut être utilisé pour créer un arc où le lien étendu serait le point de départ ou d'arrivée (le lien étendu, comme n'importe quel autre élément d'un document XML, peut être considéré comme une ressource composant un autre lien)
<b>xlink:title</b>	une chaîne de caractères	donne un commentaire sur le lien étendu dans son ensemble

XML - 08/02/2002

141

## Liens étendus (2)

attribut	valeur	signification
<b>xlink:type</b> (obligatoire)	locator	définition d'un élément englobé dans un lien étendu et servant à définir un point d'attache pour les arcs du lien (permettra de définir l'emplacement du point de départ ou d'arrivée d'un ou plusieurs arcs)
<b>xlink:href</b> (obligatoire)	une URL	<a href="#">emplacement d'une ressource externe, peut désigner un fragment d'un document XML. (c.f. la spécification de Xpointer)</a>
<b>xlink:role</b>	identificateur (avec préfixe si l'identificateur est défini dans un autre fichier)	l'identificateur utilisé ici peut être utilisé comme valeur de l'attribut xlink:from ou xlink:to d'un autre élément pour créer un arc entre l'élément courant et cet autre élément
<b>xlink:title</b>	une chaîne de caractères	donne un commentaire sur la ressource externe désignée par l'élément courant

XML - 08/02/2002

142

## Liens étendus (3)

attribut	valeur	signification
<b>xlink:type</b> (obligatoire)	arc	établissement d'un arc entre deux des ressources associées au lien étendu, l'une constituant le point de départ de l'arc et l'autre le point d'arrivée.
<b>xlink:role</b>	identificateur (avec préfixe si l'identificateur est défini dans un autre fichier)	l'identificateur utilisé ici peut être utilisé comme valeur de l'attribut xlink:from ou xlink:to d'un autre élément pour créer un arc entre l'élément courant et cet autre élément.
<b>xlink:from</b>	identificateur (avec préfixe si l'identificateur est défini dans un autre fichier)	l'identificateur utilisé ici doit être défini dans un autre élément à l'aide de l'attribut xlink:role. Il indique alors que cet autre élément est le point de départ de l'arc spécifié par l'élément courant.
<b>xlink:to</b>	identificateur (avec préfixe si l'identificateur est défini dans un autre fichier)	l'identificateur utilisé ici doit être défini dans un autre élément à l'aide de l'attribut xlink:role. Il indique alors que cet autre élément est le point d'arrivée (destination) de l'arc spécifié par l'élément courant.
<b>xlink:show</b>	new, replace, embed, undefined	comme pour les liens simples
<b>xlink:actuate</b>	onLoad, onRequest, undefined	comme pour les liens simples

XML - 08/02/2002

143

## Liens étendus (4)

attribut	valeur	signification
<b>xlink:type</b>	resource	définition d'une ressource locale englobée dans le lien étendu
<b>xlink:role</b>	identificateur (avec préfixe si l'identificateur est défini dans un autre fichier)	l'identificateur utilisé ici peut être utilisé comme valeur de l'attribut xlink:from ou xlink:to d'un autre élément pour créer un arc entre l'élément courant et cet autre élément
<b>xlink:title</b>	une chaîne de caractères	le texte fourni pourra servir d'indice supplémentaire, par exemple en étant affiché quand la souris se trouve sur l'élément contenant le lien

XML - 08/02/2002

144



## Liens étendus - exemple

```
<image xlink:type="extended" xlink:title="informations disponibles" >
<html:img src="..." xlink:type="resource" xlink:role="depart" />
<res xlink:type="locator" xlink:href="..." xlink:role="photographe" />
<res xlink:type="locator" xlink:href="DetailsPhoto382.xml#DatePrise" xlink:role="date" />
<res xlink:type="locator" xlink:href="Photo382Big.jpg" xlink:role="agrandissement" />
<res xlink:type="locator" xlink:href="Copyright.xml" xlink:role="copyright" />
<vers xlink:type="arc" xlink:from="depart" xlink:to="photographe" xlink:title="prise par"
xlink:show="new" xlink:actuate="onRequest" />
<vers xlink:type="arc" xlink:from="depart" xlink:to="date" xlink:title="date de la prise"
xlink:show="embed" xlink:actuate="onLoad" />
<vers xlink:type="arc" xlink:from="depart" xlink:to="copyright" xlink:title="&#169;"
xlink:show="replace" xlink:actuate="onRequest" />
<vers xlink:type="arc" xlink:from="depart" xlink:to="agrandissement" xlink:title="version
1280x1024 pixels" xlink:show="replace" xlink:actuate="onRequest" />
</image>
```

XML - 08/02/2002

145

## Référence croisée avec des liens étendus

```
<?xml version="1.0"?>
<doc>
<argument xlink:type="extended">
<theme xlink:type="locator" href="idea1">
<theme xlink:type="locator" href="idea2">
</argument>
<para id="idea1">A first impression of the subject matter
would inevitably lead to the conclusion that ... </para>
<para id="idea2">Contrary to our earlier conclusion, it
would appear that the evidence points to a different ...
</para>
</doc>
```

XML - 08/02/2002

146

## Xpointer Adressage avancé en XML

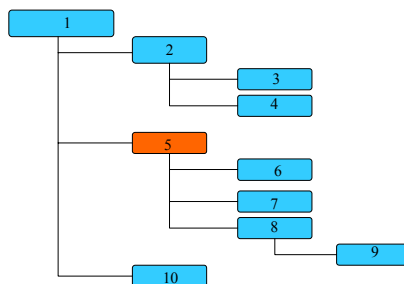
Deux types de termes de localisation :

- Absolus
  - root, origin, id(name), html(name-value)
- Relatifs
  - child, descendant, ancestor, preceding, following, psibling, fsibling

XML - 08/02/2002

147

## Exemple



XML - 08/02/2002

148

## Points essentiels

- I. Présentation de XML
- II. Anatomie d'un document XML / Visualisation
- III. Feuilles de style
- IV. Vérification d'un document XML
  - document bien-formé
  - validation : DTD / Schéma
- V. Liens XML
- VI. Traitement de XML
- VII. XML Schemas
- VIII. Conclusion et références

XML - 08/02/2002

149

## VI. Traitement de XML

XML - 08/02/2002

150

## Pourquoi traiter du XML ?

- Distribution à plusieurs médias
- Distribution à différents groupes
- Ajout, suppression et restructuration d'information
- Chargement dans une base de données
- ...

XML - 08/02/2002

151

## Distribution à plusieurs médias

- Sur le web :  
Transformation du XML au vol pour s'adapter à la bonne version de HTML (3.2, 4, DHTML ...)
- En hard copy :  
conversion de XML en RTF (Rich Text Format) pour utiliser avec MS Word, ou un autre langage de markup
- Sur CD-ROM
- Fichiers d'aide
- ...

XML - 08/02/2002

152

## Distribution à différents groupes

- Débutants, utilisateurs avancés, experts
- Membres, non membres
- ...

XML - 08/02/2002

153

## Exemple : en fonction de l'expérience de l'utilisateur

```
<?xml version="1.0"?>
<procedure type="disassemble">
  <object>
    <type>Mismatcher</type>
    <seriesnr>21568</seriesnr>
  </object>
  <title>How to disassemble Mismatcher
  21568</title>
  <steps experience="firsttime">
    <step>
      <action>Press ....
      </action>
      <result>The door ....
      </result>
    </step>
    <step>
      <action>Push ....
      </action>
      <result>The back ...
      </result>
    </step>
  </steps>
  <steps>
    <action>Drill ....
    </action>
    <result>Part x123 ...
    </result>
  </steps>
  <step experience="donebefore">
    <action>Throw it on the ground.
    </action>
  </step>
  </steps>
  <tip>Take a bottle of ...
  </tip>
</procedure>
```

XML - 08/02/2002

154

## Exemple de traitement

- Utilisateur débutant :  
→ Supprimer les éléments *steps* dont l'attribut *experience* vaut *donebefore*
- Utilisateur avancé :  
→ Supprimer les éléments *tip*

XML - 08/02/2002

155

## Trois paradigmes de traitement

Document XML :

- Fichier texte
- Séquence d'événements : "event-based"
- Hiérarchie : "tree-based"

XML - 08/02/2002

156

## VI.1. XML vu comme un fichier texte

- Mélange de données et de markup
- Outils de manipulation de texte utilisant des expressions régulières
  - ➔ Grep, perl, python
- Ex : nombre de musiciens dans le fichier musicians.xml
 

```
grep -c "<musician>" musicians.xml
```

XML - 08/02/2002

157

## VI.2. XML vu comme une séquence d'événements

- Début du document
  - Lecture de balises de début et de fin
  - Rencontre de commentaires
  - Processing Instructions
  - ...
- ➔ Event handler : code exécuté lorsqu'un événement se produit

XML - 08/02/2002

158

## Événements générés par musicians.xml

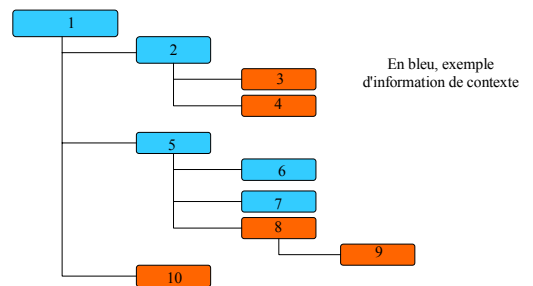
```
<?xml version="1.0" ?>
<musicians>
<musician>
<name>Joey Baron
</name>
<instrument>drums
</instrument>
<NrOfRecordings>1
</NrOfRecordings>
</musician>
<musician>
<name>Bill Frisell
</name>
<instrument>guitar
</instrument>
<NrOfRecordings>3
</NrOfRecordings>
</musician>
</musicians>
```

```
Start document
Start element: musicians
Start element: musician
Start element: name
Characters: Joey Baron
End element: name
Start element: instrument
Characters: drums
End element: instrument
Start element: NrOfRecordings
Characters: 1
End element: NrOfRecordings
End element: musician
...
End element: musicians
End document
```

XML - 08/02/2002

159

## Parcours du document XML



XML - 08/02/2002

160

## Avantages / inconvénients du traitement par événements

- Avantages :
  - Simple
  - Rapide
  - Consomme peu de mémoire
- Inconvénients :
  - Impossible de regarder en avant
- Implémentations :
  - Omnimark
    - <http://www.omnimark.com/develop/>
  - SAX (Simple API for XML)
    - <http://www.megginson.com/SAX/index.html>

XML - 08/02/2002

161

## VI.3. XML vu comme une hiérarchie / un arbre

Un processeur "tree-based" traduit le document XML en une structure d'arbre interne et permet à une application de naviguer dans cet arbre

2 passages :

- 1er passage : analyse et construction de l'arbre
  - 2ème passage : traitement des données
- Accès au document complet
    - ➔ Accès à toute l'information

XML - 08/02/2002

162

## Avantages / inconvénients du traitement "tree-based"

- Avantages :
  - Accès à tout le document
  - ➔ On peut regarder en avant
- Inconvénients :
  - Plus difficile de d'abord construire un arbre, puis de naviguer à l'intérieur
  - Nécessite beaucoup de mémoire
  - Plus lent
  - Nécessite deux passages
- Implémentation :
  - DOM (Document Object Model)  
<http://www.w3.org/TR/DOM-Level-2-Core/>

XML - 08/02/2002

163

## Programmation avec le Document Object Model

- Interface indépendante de la plate-forme et du langage, qui permet aux programmes et aux scripts d'accéder et de mettre à jour le contenu, la structure et le style des documents de manière standardisée
- DOM : ensemble d'interfaces et d'objets pour gérer des documents HTML et XML

XML - 08/02/2002

164

## Utilisation du DOM

- Utilisation de JavaScript ou VBScript dans les pages Web
- Application externe accédant au document par le browser
- Parser externe implémentant le DOM

XML - 08/02/2002

165

## Exemples d'objets et méthodes du DOM

- Objets :  
Document, Node, NodeList, Attr, Element, Text, ...
- Méthodes :  
insertBefore, replaceChild, createElement, getElementByTagName, createTextNode, ...

XML - 08/02/2002

166

**www.Mcours.com**

Site N°1 des Cours et Exercices

Email: [mymcours@gmail.com](mailto:mymcours@gmail.com)

## Points essentiels

- I. Présentation de XML
- II. Anatomie d'un document XML / Visualisation
- III. Feuilles de style
- IV. Vérification d'un document XML
  - document bien-formé
  - validation : DTD / Schéma
- V. Liens XML
- VI. Traitement de XML
- VII. XML Schemas
- VIII. Conclusion et références

XML - 08/02/2002

167

## VII. Schémas XML

XML - 08/02/2002

168

## Schémas XML

- DTD parfois inadaptées
  - Peu de possibilités de vérifier les données à l'intérieur d'un élément
  - La DTD n'est pas un document XML ni SGML, elle a une syntaxe particulière
    - Apprentissage plus long
    - Ne bénéficie pas de l'automatisation
- Besoin d'un schéma XML qui permette de mieux contrôler les données tout en partageant la syntaxe XML

XML - 08/02/2002

169

## Schémas XML

Un schéma :

- Spécifie la syntaxe d'une classe de documents XML
- Définit les éléments et attributs, ainsi que les contraintes sur les valeurs de ceux-ci
- Est contenu dans un élément du genre :

```
<xsd:schema xmlns:xsd=http://www.w3.org/1999/XMLSchema>
</xsd:schema>
```

XML - 08/02/2002

170

## Élément à contenu simple

- Contenu : chaîne de caractères, nombre, ...
  - ➔ Définition avec un élément vide

```
<xsd:element name="auteur" type="xsd:string"
minOccurs="1" />
```

XML - 08/02/2002

171

## Élément contenant d'autres éléments

```
<xsd:element name="article" >
<xsd:complexType content="elementOnly">
<xsd:element name="titre" type="xsd:string"
minOccurs="1" />
<xsd:element name="auteur" type="xsd:string"
minOccurs="1" />
<xsd:element name="date" type="xsd:string"
minOccurs="1" />
<xsd:element name="lieu" type="xsd:string"
minOccurs="1" />
</xsd:complexType>
</xsd:element>
```

XML - 08/02/2002

172

## Attributs minOccurs et maxOccurs

- Nombre minimum et maximum de sous-éléments autorisés dans le contexte de l'élément englobant
- Si ce n'est pas indiqué explicitement :
  - maxOccurs vaut 1 quand minOccurs vaut 0
  - maxOccurs vaut minOccurs sinon
- Si ce n'est pas indiqué explicitement :
  - minOccurs vaut 1

XML - 08/02/2002

173

## Élément xsd:complexType

- Définit le contenu de l'élément qui le contient
- Peut avoir l'attribut content, qui indique le type de contenu que l'élément peut avoir :
  - elementOnly
  - textOnly
  - Mixed
  - empty

XML - 08/02/2002

174

## Exemple

```
<xsd:element name="article" >
  <xsd:complexType content="elementOnly">
    ...
    <xsd:element name="texte" >
      <xsd:complexType content="mixed" >
        <xsd:element name="grand" type="xsd:string" minOccurs="0"
maxOccurs="unbounded" />
        <xsd:element name="image" minOccurs="0" maxOccurs="unbounded"
xmlns:html="http://www.w3.org/Profiles/XHTML-transitional" >
          <xsd:complexType content="elementOnly" >
            <xsd:element ref="html:img" minOccurs="1" />
            <xsd:element name="legende" type="xsd:string" minOccurs="0" />
          </xsd:complexType>
        </xsd:element>
      </xsd:complexType>
    </xsd:element>
  </xsd:complexType>
</xsd:element>
```

XML - 08/02/2002

175

## Définition d'attributs

■ Exemple :

```
<article titre="Un journaliste accuse, un policier dément" auteur="Alain Connu"
date="14 juin 1972" lieu="banquise">
```

■ Schéma correspondant :

```
<xsd:element name="article" >
  <xsd:complexType content="elementOnly">
    ... <!-- les attributs doivent être déclaré après les sous-éléments -->
    <xsd:attribute name="titre" type="xsd:string" minOccurs="1" />
    <xsd:attribute name="auteur" type="xsd:string" use="required" />
    <xsd:attribute name="date" type="xsd:date" use="required" />
    <xsd:attribute name="lieu" type="xsd:string" use="required" />
  </xsd:complexType>
</xsd:element>
```

XML - 08/02/2002

176

## Valeur de l'attribut use

- prohibited : l'attribut ne doit pas apparaître
- optional : peut apparaître ou non
- required : doit apparaître
- fixed : ne peut prendre qu'une seule valeur
- default : peut ne pas être défini, auquel cas il prendra une valeur par défaut

XML - 08/02/2002

177

## Référence au schéma

■ <ElementRacine xmlns="UnFichier.xsd">  
...  
</ElementRacine>

■ ElementRacine xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"  
xsi:schemaLocation="UnURISymbolisantL.EspaceDeNom UnFichier.xsd">  
...  
</ElementRacine>

■ <ElementRacine xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="UnFichier.xsd">  
...  
</ElementRacine>

XML - 08/02/2002

178

## Points essentiels

- I. Présentation de XML
- II. Anatomie d'un document XML / Visualisation
- III. Feuilles de style
- IV. Vérification d'un document XML
  - document bien-formé
  - validation : DTD / Schéma
- V. Liens XML
- VI. Traitement de XML
- VII. XML Schemas
- VIII. Conclusion et références

XML - 08/02/2002

179

## VIII. Conclusion

- L'usage d'XML va largement déborder le WWW
- XML va devenir rapidement la *lingua franca* de l'informatique, parlée autant par les SGBD que par les outils de bureautique et de documentation, par les logiciels de gestion aussi bien que par les applications techniques et scientifiques
- Automatisation des activités administratives et logistiques
- Simplification de l'Échange de Données Informatisé (EDI)

XML - 08/02/2002

180

## Conclusion (suite)

On n'a pas fini d'entendre parler d'XML !

## Références

- <http://www.w3.org/>
- Spécification en français  
[http://babel.alis.com/web\\_ml/xml/REC-xml.fr.html](http://babel.alis.com/web_ml/xml/REC-xml.fr.html)
- The XML handbook  
Charles F. Goldfarb and Paul Prescod  
ISBN 0-13-081152-1
- <http://www.oasis-open.org/cover/xml.html>

***www.Mcours.com***

Site N°1 des Cours et Exercices

Email: [mymcours@gmail.com](mailto:mymcours@gmail.com)

## Questions ?