



Technologie XML

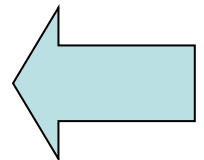


Plan

- Introduction
- La syntaxe XML
- Les schémas
 - DTD (Document Type Definition)
 - XSD (XML Schema Definition)
- Les espaces de noms
- Les feuilles de style
 - CSS
 - XSL (XML StyleSheet Language)
- XPath
- XQuery
- Autres standards
- Parseurs : DOM et SAX

Introduction

- XML, C'est quoi ?
- Domaine d'application
- Outils logiciels



C'est Quoi?

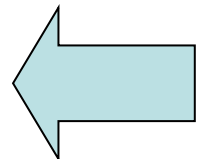
- XML : eXtensible Markup Language

- C'est un langage de Marquage

- IL est extensible

- Très strict

- Met l'accent sur la structure des documents





Langage de Marquage

- **Langage de Marquage = langage se basant sur la notion de balisage**
- **Marquage ou Balisage = du code inséré dans un document texte. IL stocke les informations nécessaire pour le traitement du document**



Markup Language

- **Exemples**

- **Marquage procédural**

- **Marquage générique**

- **SGML : Standard General Markup Lang**

SGML

- C'est un standard de l'ISO
- C'est un métalangage : il permet de définir d'autres langages de balisage
 - HTML est une application de SGML

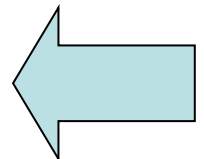


HTML : le succès du Web

- C'est un langage destiné spécialement pour le web
- Très simple,
- Très tolérant
- Derrière le succès du web

HTML : limité !

- N'est plus un petit langage
 - ◆ Un nombre énorme de balise
 - ◆ Autres langages complémentaires
- IL est malgré tout très limité
 - ◆ IL ne répond pas à tous les besoins
 - ◆ IL mélange le contenu, la structure et la présentation



Extensibilité

- Les applications Web ont de plus en plus de nouveaux besoins et bien spécifiques:
 - Commerce : balises pour les produits, les prix, les clients...
 - Moteurs de recherches : balises pour les mots clés, pour la description,...
 - Mathématiques : balises pour les formules, les théorèmes, ...
 - ...



HTML

- Ajouter de nouvelles balises ne résoud pas le problème: on aura toujours besoin d'autres
- De plus : certaines applications préfèrent un petit langage plutôt qu'un grand langage (par exemple: l'accès au web par les téléphones mobiles)



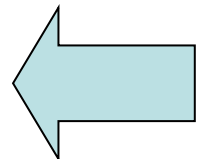
Solution ?

- **Comment donc obtenir un langage avec peu de balise et qui répond à tous les besoins?**

L'extensibilité

Extensibilité

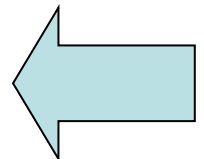
- C'est la capacité de créer son propre jeu de balises
- XML: pas de balises prédéfinies
- XML est donc un métalangage : Vous pouvez créer vos propres langages de balisages



XML est Très strict

En effet :

- La tolérance de HTML à un prix:
 - Des navigateurs énormes et coûteux
- La tolérance de HTML n'est plus très importante:
 - Les utilisateurs utilisent de plus en plus les éditeurs graphiques





XML et les document structuré

- Exemple...



XML et les documents structurés

- XML est un « ensemble de standards » pour échanger et publier l'information d'une manière structurée.
- XML est utilisé pour décrire et manipuler les documents structurés

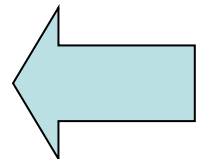


XML et les documents structurés

- XML structure les document sous forme d'arborescence , mais il n'impose pas comment peupler cette arborescence : il est donc très flexible
- XML fournit un mécanisme pour manipuler à la fois l'information et sa structure sous-jacente.

XML et les documents structurés

- XML offre des mécanisme pour manipuler **automatiquement** l'information par des application.
- Cette manipulation passe bien sûr à travers la structure



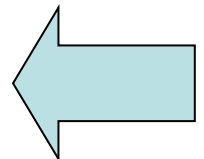


Application de XML

- **Application de documents**
 - **Manipulation des informations destinées pour les humains**
- **Application de données**
 - **Manipulation des informations destinées pour les logiciels**

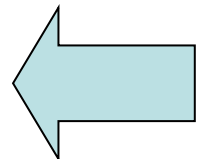
Domaines d'application

- Publication
- Commerce électronique
- Gestion électronique du contenu
- Échange de données entre Systèmes d'informations hétérogènes
- ...



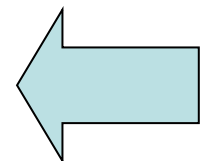
Les Outils Logiciels

- **Navigateurs XML**
- **Éditeurs XML**
- **Parseurs XML**
- **Processeurs XSL**



La syntaxe XML

- Éléments
 - Les noms en XML
 - La casse
 - Les attributs
 - Les éléments vides
- Les chevauchements
- L'élément racine
- Les commentaires
- La déclaration XML
- Les entités



Les éléments XML

● **<balise>** Contenu **</balise>**

Les noms XML

- Les noms des éléments doivent commencer par : **a, b, c,...** , **A, B, C,...** ou **_**.
- Le reste du nom est constitué par des **lettres**, des **chiffres**, **_** , **-** , **.** , **:** .
- Les nom en XML ne doivent pas commencer par **xml**
- Les **espaces** ne sont pas autorisés
- Le «**:**» a une signification particulière. Éviter de l'utiliser

La casse

- La casse est importante dans les noms XML
 - <ETUDIANT>
 - <etudiant>
 - <EtuDiant>
- Par convention, on préfère le minuscule pour les noms des éléments et des attributs

Attributs

- Pour ajouter une information supplémentaire aux éléments

<élément **attribut** = "valeur" >

Contenu

</élément>

- Les mêmes règles de nommage que les éléments




Parseurs

A. Zinedine



DOM

- **Node**
- **NodeList**
- **NamedNodeMap**
- **DOMImplementation**
- **DOMException**

- 
- Document (9)
 - ProcessingInstruction (7)
 - DocumentType (10)
 - DocumentFragment (11)
 - Element (1)
 - Attr (2)
 - CaracterData
 - Text (3)
 - Comment (8)
 - CDATASection (4)
 - Entity (6)
 - EntityReference (5)
 - Notation (12)



Node

- nodeName
- NodeType
- nodeValue
- parentNode
- childNodes
- firstChild
- lastChild
- previousSibling
- nextSibling
- attributes



Document

- **documentElement** (racine)
- **docType**



Document

- **CreateAttribute(name)** [Attr Obj]
- **CreateCDATASection(data)**
- **createComment(data)**
- **createElement(name)**
- **createEntityReference(name)**
- **createProcessingInstruction(target,data)**
- **createTextNode(data)**
- **createDocumentFragment()** (empty)



Node

- **appendChild(child)**
- **insertBefore(child,before)**
- **replaceChild(child,toReplace)**
- **removeChild(child)**
- **cloneNode(deep)**
- **hasChildNodes()**



Element

- `setAttribute(name,value)`
- `getAttribute(name)`
- `removeAttribute(name)`

- `setAttribute(Attr)` [Attr Obj]
- `setAttributeNode(attr)`
- `getAttributeNode(attr)`
- `removeAttributeNode(node)`



CharacterData

- `appendData(data)`
- `insertData(offset,data)`
- `deleteData(offset,length)`
- `replaceData(offset,length,data)`
- `substringData(offset,length)`



N'est pas encore disponible

A. Zinedine

