

XML

Technologies Internet

2012

Introduction

Langages de marquage

HTML

XML : Qu'est ce que c'est ?

XML : Balises & Entités

XML : Structuration

XML : Well Formed

Langages de marquage

Langage de marquage (Markup Language)

- Un langage de marquage est un langage qui définit les règles de marquage qui donnent une signification à la structure et au contenu des documents.
- Ces règles définissent la grammaire et la syntaxe du langage.
- Exemples de langages de marquage :
 - HTML
 - XML

Langage de marquage

- Un langage de marquage est composé :
 - Éléments
 - Attributs
 - Entités

HTML

HTML: ELEMENTS

- Les éléments sont les blocs de données principaux des documents XML et HTML.
- Exemples d'éléments HTML: "body" et "table".
- Exemples d'éléments XML: "livre" et "isbn".
- Les éléments peuvent contenir du texte, d'autres éléments, ou être vides.
- Exemples d'éléments HTML vides: "hr", "br" et "img".

HTML : ATTRIBUTS

- Les attributs fournissent de l'information additionnelle concernant les éléments.

- Exemple :

```
<IMG SRC="ordinateur.gif" />
```

```
<nourriture cholesterol="30mg">frites</nourriture>
```


HTML : ENTITES

- Les entités sont des abréviations.
- Exemples :
`<` est l'abréviation prédéfinies du caractère <

HTML : FAIBLESSE & LIMITES

- HTML a un ensemble pré-déterminé de balises
 - Il n'est pas possible de créer de nouvelles balises.
- HTML est un langage de présentation
 - Les balises donnent des indications sur la manière de présenter et non sur le contenu.

XML

XML : Définition

- XML est un langage de marquage qui permet de décrire le contenu des éléments.
- XML est l'abréviation eXtensible Markup Language
- C'est un sous-ensemble de SGML (Standard Generalized Markup Language)
- Il est spécialement adapté pour les échanges sur le Web
- Avec XML, l'utilisateur peut créer ses propres balises qui permettent de décrire le contenu de ce qui est "marqué".

XML : Utilisations

- XML s'est imposé comme le format standard de communication entre applications.
 - ▶ EAI / ESB
 - ▶ Web Services (SOAP)...
 - ▶ BPMN / BPEL
- XML est utilisé pour la configuration :
 - ▶ Fichiers de configurations de logiciels (remplace les **.ini**)
 - ▶ Fichiers de configuration de serveurs d'applications (Tomcat : **server.xml** , ...)
 - ▶ L'intégration : LDAP, ESB, EAI
- XML est utilisé aussi pour les bases de données
 - ➔ Fichiers XML contiennent la structure et la donnée

XML : Balises & Éléments

- Tous les éléments doivent avoir une balise d'ouverture et une balise de fermeture.

Exemple :

```
<LIVRE> . . . </LIVRE>
```

- La seule exception est l'élément vide (Empty element) dans lequel il n'y a pas de contenu d'élément et qui combine la balise ouvrante et fermante.

Exemple :

```
<prix Canada="$70.99"/>
```

XML : Attributs

- Les attributs sont à l'intérieur de la balise ouvrante d'un élément.
- Syntaxe :
attribut="valeur"
- Exemple :
<LIVRE SUJET="XML">.
L'attribut SUJET de l'élément LIVRE a la valeur "XML".
- En XML, les valeurs doivent toujours être encadrées par des guillemets.

XML : Entités

- Il existe deux catégories d'entités:
 - Entités générales
 - Entités de paramètre

XML : Entités Générales

- Les entités générales sont des variables utilisées pour définir des chaînes de caractères.
- Elles sont utilisées pour ne pas avoir à taper de longues chaînes de caractères plusieurs fois dans un document.
- Exemple :
 &xml; pour "Extensible Markup Language"
- Note :
 Si la chaîne de caractères doit être modifiée, on la modifie seulement une fois.

XML : Entités de paramètres

- Les entités de paramètre sont pré-définies dans XML

Entité de paramètre	Caractère
<	<
>	>
&	&
"	"
'	'

XML : Données Textuelles

- Le texte entre les balises d'un élément est appelé données textuelles (character data).
- Une donnée textuelle est composée de n'importe quel caractère à l'exception du caractère <.
- Note: on utilise l'entité < à la place du caractère <.
- Le nom d'une balise peut commencer par une lettre ou le symbole souligné _ suivi de n'importe quelle combinaison de lettres, de chiffres, de tiret (-), de souligné (_) ou de point (.).
- La seule exception est que le nom ne peut pas débiter par xml en miniscules.

XML : Commentaires & Sections ignorés

- Le texte entre les balises d'un élément est appelé données textuelles (character data).
- Une donnée textuelle est composée de n'importe quel caractère à l'exception du caractère <.
- Note: on utilise l'entité < à la place du caractère <.
- Le nom d'une balise peut commencer par une lettre ou le symbole souligné _ suivi de n'importe quelle combinaison de lettres, de chiffres, de tiret (-), de souligné (_) ou de point (.).
- La seule exception est que le nom ne peut pas débiter par xml en miniscules.

XML : Commentaires & Sections ignorés

- o Commentaires

`<!-- courses du club -->`

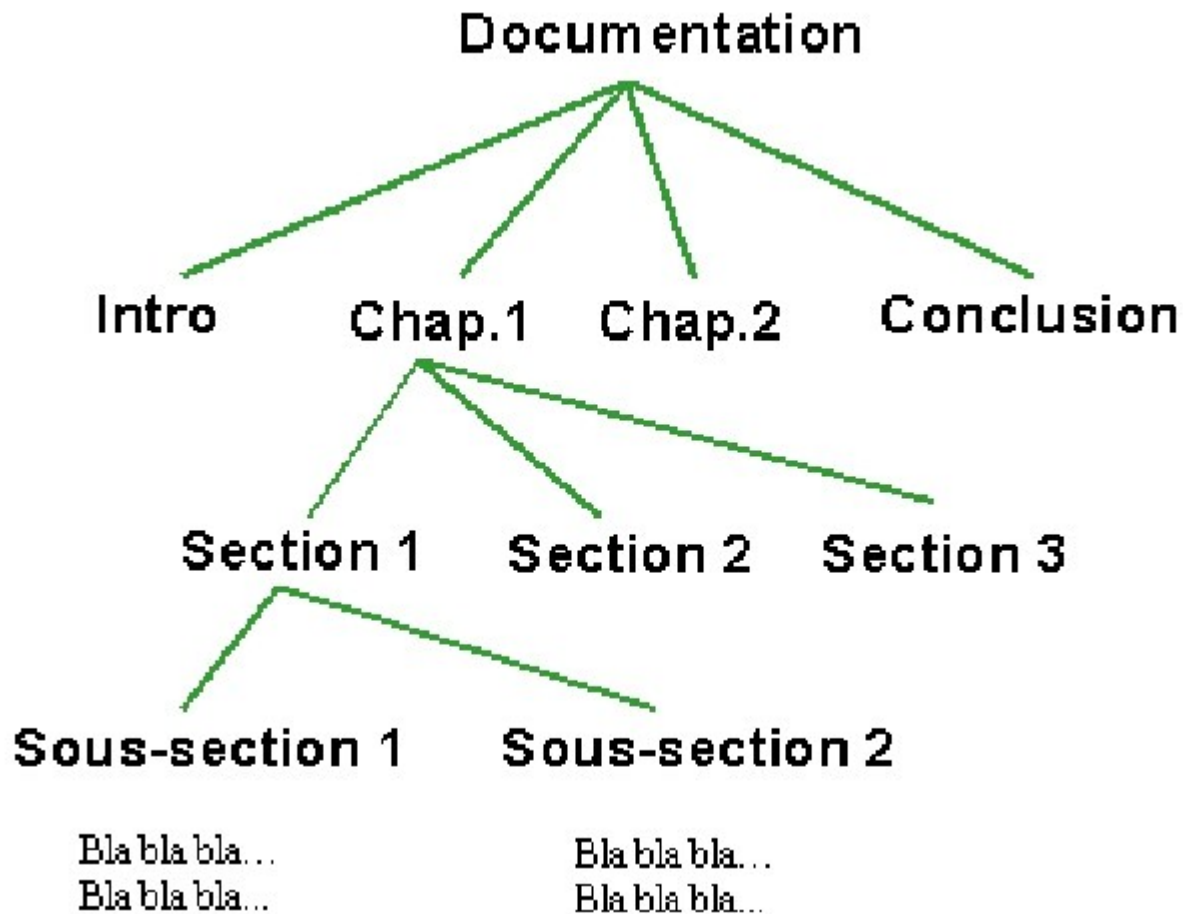
- o Instructions de traitement

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

- o Sections ignorées

`<![CDATA[10,100,20,200]]>`

XML : Structure d'un document



```
<Documentation>
  <intro>
  </intro>
  <Chap1>
    <Section1>
      <Sous-section1>
        Bla bla bla
      </Sous-section1>
      <Sous-section2>
        Bla bla bla
      </Sous-section2>
    </Section1>
    <Section2>
    </Section2>
    <Section3>
    </Section3>
  </Chap1>
  <Chap2>
  </Chap2>
  <Conclusion>
  </Conclusion>
</Documentation>
```

XML : Fichier XML

- Un fichier XML a pour extension .xml
- XML est sensible à la casse. (case-sensitive)
- Les espaces blancs ne sont pas ignorés.
- La première ligne du fichier contient l'indication qu'il s'agit d'un document XML:

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

Il est à noter que xml est en minuscule et qu'il n'y a pas d'espace blanc entre l'ouverture de la balise (<) et le point d'interrogation (?)

XML : Exemple de fichier *livres.xml*

```
<?xml version="1.0"                "ISO-8859-1" ?>
<livres>
<livre>
  <titre>HTML 4, XML et Java 2</titre>
  <isbn>2-7440-0628-9</isbn>
  <auteurs>
    <nom_auteur>Eric Ladd</nom_auteur>
    <nom_auteur>Jim O'Donnel</nom_auteur>
  </auteurs>
  <description>
    Ressources d'Experts HTML 4, XML et Java 2 se
    révèle l'ouvrage le plus complet .
  </description>
  <prix Can="$70.99"/>
</livre>
</livres>
```


XML : Fichier XML / Structuration

- Tous les documents XML doivent avoir une balise UNIQUE d'ouverture et de fermeture.

Exemple :

Dans le fichier livres.xml,
cette balise est <livres>

- L'élément correspondant est appelé l'élément racine (root element)
- Tous les autres éléments sont contenus à l'intérieur des balises de l'élément racine; ce sont les sous-éléments (child elements).

```
<?xml version="1.0"          "ISO-8859-1" ?>
<livres>
  <livre>
    <titre>HTML 4, XML et Java 2</titre>
    <isbn>2-7440-0628-9</isbn>
    <auteurs>
      <nom_auteur>Eric Ladd</nom_auteur>
      <nom_auteur>Jim O'Donnel</nom_auteur>
    </auteurs>
    <description>
      Ressources d'Experts HTML 4, XML et Java 2 se
      révèle l'ouvrage le plus complet .
    </description>
    <prix Can="$70.99"/>
  </livre>
</livres>
```

XML : Fichier XML / Structuration

- Entre les balises `<livres>` et `</livres>` nous trouvons
 - ▶ les balises **`<livre>`** **`</livre>`** qui vont contenir les balises concernant les détails d'un livre comme le titre, le code ISBN, les auteurs, une description et le prix.
 - ▶ Pour le titre, la balise **`<titre>`**
 - ▶ Pour le code ISBN, la balise **`<isbn>`**
 - ▶ Comme il y a plusieurs auteurs au livre, la balise `<auteurs>` contient les balises **`<nom_auteur>`**
 - ▶ Pour la description, la balise **`<description>`**
 - ▶ Pour le prix, on utilise un élément vide (empty element); à noter que le montant est contenu dans l'attribut `Can`

XML : Structure Type

- Déclaration XML
`<?xml version="1.0"?>`
- Élément racine
`<livres> . . . </livres>`

Éléments

éléments, éléments vides, attributs, commentaires, entités, instructions

`<CHAPITRE>`

`<!-- Nouvelle section -->`

`
<TITRE>&xml; (XML)<TITRE>`

`<SECTION NUMERO="2.1">`

XML : bien formé

- Un document XML est dit **bien formé (Well-Formed)** s'il respecte les règles suivantes:

- ▶ Le document doit contenir au moins un élément.
- ▶ Il doit contenir un élément racine (Root Element).
- ▶ Tous les autres éléments doivent être contenus à l'intérieur des balises de l'élément racine et ils doivent être imbriqués correctement.
- ▶ Le document doit respecter les règles de syntaxe XML.

XML : Well Formed

```
<?xml version="1.0"?>
<inventaire>
  <contenants>
    <contenant>
      <marque>Dominion</marque>
      <capacite>3 litres</capacite>
    </contenant>
    <contenant>
      <marque>Atlas</marque>
    </contenant>
  </contenants>
</inventaire>
```