


RIHAL Grégroy Licence Multimedia



XML

Le langage du futur

Sommaire

1. Informations sur le XML	Page 2 - 3
a. Prérequis	
b. Le XML en lui même ne fait rien	
c. Les limites et les objectifs	
2. Le langage xml	Page 4
a. Le XML	
b. L'avenir du XML	
3. La syntaxe du XML	Page 5 - 7
4. Le premier document XML	Page 8 - 9
a. La syntaxe	
b. Elaboration du fichier	
c. Résultat	
5. Le DTD	Page 10 - 11
a. Le DTD interne	
b. Le DTD externe	
c. Valide	
6. XML et CSS	Page 12 - 13
a. Le CSS	
b. Exemple	
7. XML et XSL	Page 14 - 15
a. Le XSL	
b. Exemple	
8. Afficher du XML dans du HTML	Page 16 - 17
a. Comment faire?	
b. Exemple	
9. Les éditeurs XML	Page 18
10. Les livres	Page 19

Informations sur le XML

a. Prérequis

Si le langage Html est accessible au plus grand nombre, avec le langage XML vous jouez déjà un peu dans «la cour des grands». Le XML est de loin plus abstrait et donc plus complexe que le Html. Bien que cette documentation se limitera à une découverte basique du XML, il est quasi indispensable pour en tirer quelques profits d'avoir :

- une connaissance et une pratique aigüe du langage Html.
- une connaissance et une pratique de la conception de pages Web.
- de bonnes notions de feuilles de style (CSS).
- des notions de Javascript ou de VBscript.

b. Le XML en lui même ne fait rien

Alors que le Html a été conçu pour afficher de l'information, le XML a été créé pour structurer de l'information. Il ne fait rien d'autre !

Voici un exemple de XML.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<demoXML>
<message>Voici du XML</message>
</demoXML>
```

Ce qui affiché dans le Internet Explorer donne le résultat suivant.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <demoXML>
  <message>Voici du XML</message>
</demoXML>
```

Le XML n'est que de l'information encodée entre des balises. Il faudra d'autres éléments, comme par exemple un fichier XSL, pour que le navigateur puisse «comprendre» vos balises et afficher ce fichier sous une forme plus conviviale.

Informations sur le XML

c. Les limites et les objectifs

Le XML est un langage de professionnels de la conception de sites et ne sera que très rarement utilisé par les amateurs, même éclairés, de la publication sur le Web. Que ces amateurs soient cependant rassurés, pour eux le Html a encore de beaux jours devant lui... Mais pour les «pros» du Web, dès qu'il s'agira de stoker, traiter, envoyer des données, le XML sera la voie informatique royale de l'avenir.

Le XML est un métalangage soit un langage pour écrire d'autres langages. Ici aussi, il n'y a que peu de chances que vous conceviez un jour votre propre langage ! Mais le XML est une véritable révolution dans le panorama des langages de publication sur le Web. Il apparaît comme incontournable car il est déjà à la base de toute une série de nouveaux langages qui sont ou qui seront utilisés dans la conception des pages Internet comme le XHTML, le successeur désigné du Html, le WML pour le Wap des téléphones mobiles, le MathML pour les mathématiques, le SOAP et à n'en pas douter bien d'autres encore. Ces nouveaux langages générés par le XML en reprennent l'esprit, les règles et la syntaxe que vous pouvez découvrir ici.

Le langage XML

a. Le XML

Le XML pour *eXtensible Markup Language* est donc un langage de balises comme le Html mais il est extensible, évolutif. En XML, les balises ne sont pas prédéfinies. C'est vous qui devez ou pouvez définir vos propres balises.

Et c'est là le problème ! Si les braves navigateurs n'avaient plus de difficultés pour afficher les balises prédéfinies du Html comme les <H1>,
 ou autres <TABLE>, que doivent-ils faire avec vos balises <ok> ou <new> ?

Le XML a comme vocation de décrire de l'information et pas d'afficher celle-ci. Ainsi le XML pourtant créé en 1999, est resté durant près de deux ans, un concept plutôt abstrait et théorique faute de moyens fiables pour en afficher le résultat. Avec le développement de nouvelles techniques comme le XSL, il est devenu possible de percevoir concrètement les énormes potentialités de ce nouveau langage.

b.L'avenir du XML

Si le Html a régné en maître sur le Web durant la dernière décennie du 20ème (1990 à 2000), le XML sera, sans aucun doute possible, le standard omniprésent pour tout ce qui concerne la manipulation et la transmission des données durant la première décennie du 21ème siècle. Mais au risque de me répéter, ce n'est pas tant le XML lui-même que vous utiliserez mais surtout les nombreux langages qui en découleront.

Seul bémol à ces prédictions euphoriques, est la relative inertie des navigateurs grands publics à permettre, courant 2001, l'exploitation de ces nouvelles prescriptions. Comme exemple on peut citer le MathML qui est un langage issu du XML et qui a pour vocation d'afficher les formules mathématiques. Ce langage bien que parfaitement défini n'est encore exploitable que par un seul navigateur, assez confidentiel, Amaya pour pas le nommer.

La syntaxe du XML

Le XML impose des règles de syntaxe très spécifiques par rapport au Html. En outre, on retrouvera ces mêmes règles de syntaxe dans tous les langages dérivés du XML comme le XHTML ou le WML par exemple.

Le XML est un langage de balises.

Mais au contraire du Html où les balises sont définies, vous devez inventer vos balises. Rappelez vous, le XML est eXtensible. Il faut donc écrire soi-même le nom des balises utilisées.

Il y a quelques règles pour la composition des noms (mais elle ne déroutent pas les habitués du Javascript) :

- Les noms peuvent contenir des lettres, des chiffres ou d'autres caractères.
- Les noms ne peuvent débuter par un nombre ou un signe de ponctuation.
- Les noms ne peuvent commencer par les lettres xml (ou XML ou Xml...).
- Les noms ne peuvent contenir des espaces.
- La longueur des noms est libre mais on conseille de rester raisonnable.
- On évitera certains signes qui pourraient selon les logiciels, prêter à confusion comme «-», «;», «.», «<», «>», etc.
- Les caractères spéciaux pour nous francophones comme é, à, ê, ï, ù sont à priori permis mais pourraient être mal interprétés par certains programmes.

On profitera de cette liberté dans les noms pour les rendre le plus descriptif possible comme par exemple <gras_et_italique>.

Les balises sont sensibles aux majuscules et minuscules [case sensitive].

Ainsi, la balise <Message> est différente de la balise <message>. La balise d'ouverture et la balise de fermeture doivent donc être identiques. Ainsi par exemple ; <Message> ... </message> est incorrect et <message> ... </message> est correct.

Une tendance se dégage pour n'écrire les balises qu'en minuscules, limitant ainsi les erreurs possibles.

Toute balise ouverte doit impérativement être fermée.

Fini les écritures bâclées du Html où l'on pouvait dans certains cas omettre la balise de fin comme pour le paragraphe <p> ou l'élément de liste .

La syntaxe du XML

Ainsi en Html, ce qui suit est affiché correctement :

```
<p>
<ul>
<li>Point 1
<li>Point 2
```

Le XML est beaucoup plus strict. On devrait avoir :

```
<p>
<ul>
<li>Point 1</li>
<li>Point 2</li>
<p>
```

Les éventuelles balises uniques ou appelées aussi balises vides, comme `
`, `<meta>` ou `` en Html, doivent également comporter un signe de fermeture soit balise/. Ainsi une balise `<meta/>` est correcte en XML.

Les balises doivent être correctement imbriquées.

Le XML étant très préoccupé par la structure des données, des balises mal imbriquées sont des fautes graves de sens.

Ainsi l'écriture suivante est incorrecte car les balises ne sont pas bien imbriquées :

```
<parent><enfant>Loïc</parent></enfant>
```

L'écriture correcte avec une bonne imbrication des éléments est :

```
<parent><enfant>Marine</enfant></parent>
```

Tout document XML doit comporter une racine.

En fait, la première paire de balises d'un document XML sera considéré comme la balise de racine [root].

Par exemple :

```
<racine>
... suite du document XML ...
</racine>
```

Si on ose faire un lien avec le Html, votre élément racine était `<body> ... </body>`. Tous les autres éléments seront imbriqués entre ces balises de racine.

La syntaxe du XML

Par exemple :

```
<parents>
  <enfants>
    <petits_enfants> ... </petits_enfants>
  </enfants>
</parents>
```

Les valeurs des attributs doivent toujours être mises entre des guillemets.

Le XML peut avoir (comme le Html) des attributs avec des valeurs. En XML, les valeurs des attributs doivent obligatoirement être entre des guillemets, au contraire du Html où leur absence n'a plus beaucoup d'importance.

Ainsi, l'écriture suivante est incorrecte car il manque les guillemets.

```
<date anniversaire=071185>
```

La bonne écriture est :

```
<date anniversaire=«071185»>
```

Le XML est un langage strict. Votre document doit impérativement respecter la syntaxe du XML. On dira alors que le document est «bien formé». Seuls les documents «bien formés» seront affichés correctement. A la moindre erreur de syntaxe, le document ne sera pas ou ne sera que partiellement affiché.

Le premier document XML

a. La syntaxe

Rien de bien compliqué mais ce document sera étoffé en cours d'étude.

<?xml version=«1.0» encoding=«ISO-8859-1»?>

La déclaration `<?xml version=«1.0»?>` indique au navigateur que ce qui suit est un document XML selon sa version 1.0. Vous remarquerez que cette balise ne comporte pas de signe de fermeture car cette balise n'est pas encore du XML. On en profite généralement pour notifier le «character set» qui indique à l'interpréteur XML le jeu de caractères à utiliser. Le jeu de caractères «ISO-8859-1» a, pour nous francophones, l'avantage d'accepter la plupart des lettres avec des accents. Mais il existe d'autres jeux de caractères comme UTF-8 ou UTF-16 plutôt destinés aux anglo-saxons car ils ne reprennent pas les accents.

<racine>

L'élément racine indispensable au XML. Vous pouvez utiliser, à votre convenance, n'importe quel nom à l'intérieur de cette balise de racine.

... suite du document XML ...

Votre document XML proprement dit, qui respectera bien entendu scrupuleusement la syntaxe du XML («bien formé»).

</racine>

Le document XML se termine obligatoirement à la fermeture de la balise de racine.

b. Elaboration du fichier

Voici un petit fichier XML.

```
<?xml version=«1.0» encoding=«iso-8859-1»?>
<racine>
<enfants>
<masculin>Loic</masculin>
<feminin>Marine</feminin>
</enfants>
</racine>
```

Le premier document XML

On le reproduit dans le programme Bloc-notes [notepad] pour les gens de Windows.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<racine>
<enfants>
<masculin>Loic</masculin>
<feminin>Marine</feminin>
</enfants>
</racine>
```

Et on l'enregistre (non pas en type de document Texte) en « Type : Tous (*.*) » sous un nom avec une extension .xml.

c. Résultat

Depuis la version 5 de Microsoft Internet Explorer, les fichiers XML s'affichent sans problèmes.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <racine>
- <enfants>
  <masculin>Loic</masculin>
  <feminin>Marine</feminin>
  </enfants>
</racine>
```

Vous remarquerez qu'il y a un petit signe - affiché devant des balises englobantes. Il suffit de cliquer sur le signe pour masquer celles-ci. Et bien entendu de cliquer sur le signe + pour les faire réapparaître.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <racine>
+ <enfants>
  </enfants>
</racine>
```

Le DTD

Le DTD ou Document Type Declaration ou encore Document Type Definition est l'ensemble des règles et des propriétés que doit suivre le document XML. Ces règles définissent généralement le nom et le contenu de chaque balise et le contexte dans lequel elles doivent exister. Cette formalisation des éléments est particulièrement utile lorsqu'on utilise de façon récurrente des balises dans un document XML.

L'étude détaillée des DTDs dépasse de loin le cadre de cet ouvrage mais un bref aperçu est cependant utile surtout pour comprendre le fonctionnement des langages dérivés du XML qui ne manquent pas d'utiliser ces fameux DTDs.

En effet, par les DTDs externes, plusieurs concepteurs peuvent se mettre d'accord pour utiliser un DTD commun pour échanger leurs données. Avec le XHTML ou le WML, vous signalez dans l'entête du document que vous utilisez (et suivez) les normes du W3C concernant les langages précités.

a. Le DTD interne

On peut inclure son propre DTD au code source du fichier XML. On parlera alors d'un DTD interne.

Le DTD interne suit la syntaxe suivante :

```
<!DOCTYPE élément-racine [  
  déclaration des éléments  
>
```

Prenons un fichier comme exemple :

<pre><?xml version="1.0" standalone="yes"?></pre>	Comme vous définissez un DTD interne, votre fichier est indépendant (standalone).
<pre><!DOCTYPE parent [<!ELEMENT parent (garcon,filles)> <!ELEMENT garcon (#PCDATA)> <!ELEMENT fille (#PCDATA)> ></pre>	Début du DTD interne avec parent comme élément de racine.
<pre><!ELEMENT parent (garcon,filles)></pre>	Cet élément racine soit parent contiendra les sous-éléments garcon et fille.
<pre><!ELEMENT garcon (#PCDATA)></pre>	#PCDATA indique au Parser XML que l'élément garcon contient des données exprimées en chiffres ou en lettres.
<pre><!ELEMENT fille (#PCDATA)></pre>	Idem pour l'élément fille.
<pre>></pre>	Fin du DTD
<pre><parent></pre>	Racine du document XML.
<pre> <garcon>Loic</garcon> <fille>Marine</fille></pre>	
<pre></parent></pre>	Fin du document XML.

Le DTD

b. Le DTD externe

Le DTD externe suivra la syntaxe suivante :

```
<!DOCTYPE élément-racine SYSTEM «nom_du_fichier.dtd»>
```

Le même fichier que ci-dessus serait alors :

```
<?xml version=«1.0» standalone=«no»?>  
<!DOCTYPE parent SYSTEM «parent.dtd»>  
<parent>  
<garcon>Loic</garcon>  
<fille>Marine</fille>  
</parent>
```

Le fichier de DTD externe (ici dans le même répertoire) «parent.dtd» contiendrait :

```
<!ELEMENT parent (garcon,fille)>  
<!ELEMENT garcon (#PCDATA)>  
<!ELEMENT fille (#PCDATA)>
```

c. Valide

Dans la littérature relative au XML, on distingue un document «bien formé» d'un document valide.

Un document valide est dit d'un document qui respecte les règles spécifiques de son DTD.

Un document «bien formé» est, pour rappel, un document qui respecte les règles générales de syntaxe du XML.

XML et CSS

a. Le CSS

Pour afficher les balises XML, on peut faire appel aux bonnes vieilles feuilles de style (CSS), maintenant classiques dans le paysage Html. A chaque balise «inventée» dans le fichier XML, on va définir un élément de style que le navigateur pourra alors afficher.

b. Exemple

A seule fin de démonstration, voici un exemple des possibilités d'une feuille de style CSS associée à un document XML.

Voici notre document XML de départ :

```
<?xml version=«1.0» encoding=«ISO-8859-1»?>
<racine>
  <enfant>
    <nom>Loïc</nom>
    <lien>garçon</lien>
    <date>07/11/83</date>
    <data>Le petit qui me dépasse d'une tête.</data>
  </enfant>
  <enfant>
    <nom>Marine</nom>
    <lien>fille</lien>
    <date>20/12/85</date>
    <data>La petite fille chérie à son papa.</data>
  </enfant>
</racine>
```

On ajoute un fichier .css dont voici le contenu :

```
<style type=«text/css»>
racine , enfant {}
nom {
  display: block;
  width: 250px;
  font-size: 16pt ;
  font-family: arial ;
  font-weight: bold;
  background-color: teal;
```

XML et CSS

```
        color: white;
        padding-left: 10px;
    }
    lien {
        display: block;
        font-size: 12pt;
        padding-left: 10px;
    }
    date {
        display: block;
        font-size: 12pt;
        color: red ;
        font-weight: bold;
        padding-left: 10px;
    }
    data {
        display: block;
        font-size: 12pt;
        font-style: italic;
        font-family: arial ;
        padding-left: 10px;
    }
</style>
```

Après avoir ajouté un lien vers le fichier css dans le fichier xml :

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

On obtient finalement :

Loïc
garçon
07/11/83
Le petit qui me dépasse d'une tête.

Marine
fille
20/12/85
La petite fille chérie à son papa.

XML et XSL

a. Le XSL

Comme le XML n'utilise pas des balises prédéfinies (car on peut inventer ses propres balises), le navigateur ne «comprend» pas les balises du XML et ne sait pas trop comment afficher un document XML.

Pour néanmoins afficher des documents XML, il est nécessaire d'avoir un mécanisme pour décrire comment le document pourrait être affiché. Un de ces mécanisme est les feuilles de style classiques du Html (CSS), mais le XSL pour eXtensible Stylesheet Language est de loin un langage de feuille de style plus adapté au XML et donc plus performant.

De façon résumée, le XSL est un langage qui transforme le XML en Html.

b.Exemple

A seule fin de démonstration, voici un exemple des possibilités du XSL associé à un document XML.

Voici notre document XML de départ :

```
<?xml version=«1.0» encoding=«ISO-8859-1»?>
<racine>
  <enfant>
    <nom>Loïc</nom>
    <lien>garçon</lien>
    <date>07/11/83</date>
    <data>Le petit qui me dépasse d'une tête.</data>
  </enfant>
  <enfant>
    <nom>Marine</nom>
    <lien>fille</lien>
    <date>20/12/85</date>
    <data>La petite fille chérie à son papa.</data>
  </enfant>
</racine>
```

XML et XSL

On ajoute un fichier .xsl dont voici le contenu :

```
<?xml version=«1.0» encoding=«ISO-8859-1»?>
<html xmlns:xsl=«http://www.w3.org/TR/WD-xsl»>
<body style=«font-family:Arial; font-size:12pt;»>
  <xsl:for-each select=«racine/enfant»>
    <div style=«background-color:teal; color:white;»>
      <span style=«font-weight:bold; color:white; padding:4px»>
        <xsl:value-of select=«nom»/></span>
        - <xsl:value-of select=«lien»/>
      </div>
      <div style=«margin-left:20px; font-size:10pt»>
        <span> Anniversaire le <xsl:value-of select=«date»/>
        </span>
        <span style=«font-style:italic»> - <xsl:value-of select=«data»/>
        </span>
      </div>
    </xsl:for-each>
  </body>
</html>
```

Après avoir ajouté un lien vers le fichier xsl dans le fichier xml :

```
<?xml:stylesheet type=«text/xsl» href=«simple.xsl»?>
```

On obtient finalement :

Loïc - garçon

Anniversaire le 07/11/83 - *Le petit qui me dépasse d'une tête.*

Marine - fille

Anniversaire le 20/12/85 - *La petite fille chérie à son papa.*

Afficher du XML dans du HTML

a. Comment faire?

On peut toujours incorporer du XML dans un fichier Html avec la balise `<xml> ... </xml>`. Mais en toute logique, quand les navigateurs rencontrent des balises incorrectes ou inconnues, rien n'est affiché. Ce sera le cas avec vos balises XML incorporées dans un fichier Html. Heureusement, on peut passer par une astuce qui répond au doux nom romantique de «îlots de données».

Les Data Islands

Derrière ce nom pour le moins bizarre, se cache une possibilité assez intéressante. Dans un fichier Html, vous pouvez créer un «îlot» de données se trouvant dans un fichier XML distinct et en extraite des données que vous pouvez alors afficher dans le document Html.

Ici, dans le fichier Html, on va désigner le fichier xml extérieur avec un identifiant id :

```
<xml id="fichierxml" src="simple.xml"></xml>
```

Dans un tableau Html, que l'on relie par un attribut à la source des données au moyen de l'identifiant désigné plus haut [`datasrc="#id"`], on peut finalement aller reprendre des données du fichier XML avec l'attribut de champ de donnée qui a comme valeur le nom de la balise XML [`datafld="balise_xml"`].

b. Exemple

Voilà toujours notre fichier XML (extérieur) :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<racine>
  <enfant>
    <nom>Loïc</nom>
    <lien>garçon</lien>
    <date>07/11/83</date>
    <data>Le petit qui me dépasse d'une tête.</data>
  </enfant>
  <enfant>
    <nom>Marine</nom>
    <lien>fille</lien>
    <date>20/12/85</date>
    <data>La petite fille chérie à son papa.</data>
  </enfant>
</racine>
```

Afficher du XML dans du HTML

Je vais créer un fichier Html classique dans lequel je voudrais reprendre des données du fichier XML et plus précisément le contenu des balises <nom>, <lien> et <date>.

```
<html>
<body>
<xml id=«fichierxml» src=«simple.xml»></xml>
<table border=«1» datasrc=«#fichierxml»>
<tr>
<td><span datafld=«nom»></span></td>
<td><span datafld=«lien»></span></td>
<td>Anniversaire le <span datafld=«date»></td>
</tr>
</table>
</body>
</html>
```

Ce qui une fois affiché, offre le résultat suivant :

Loïc	garçon	Anniversaire le 07/11/83
Marine	fille	Anniversaire le 20/12/85

Les éditeurs XML

N'attendez pas de miracle des éditeurs XML ! Comme en XML vous fabriquez sur mesure vos balises, les éditeurs ne peuvent avoir qu'un rôle d'aide à l'encodage et à la structure de votre document.

Les éditeurs XML sont cependant d'une grande utilité si vous avez de nombreuses balises récurrentes dans votre document XML. En outre, s'il en est nécessaire d'actualiser souvent les données du votre fichier Xml, il sera beaucoup plus facile de vous retrouver dans l'interface d'un éditeur que dans le fouillis de balises du code source.

Je vous ai lister les éditeurs de XML les plus connus comme :

- Microsoft XML Notepad
- XML Spy
- Xmetal
- BonFire Studio
- XMLFox
- XMLEditPro
- Peters XML Editor
- XML writer

Les livres

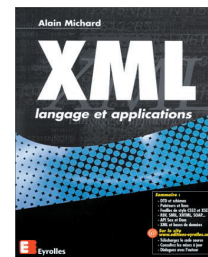
Vous souhaitez approfondir vos connaissances en XML je vous conseil d'aller acheter un de ces livres qui vous permettront de mieu appréhender le monde du xml et d'en connaitre chaque détail.



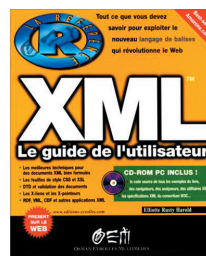
*XML et XSL
Les feuilles de styles XML*



*XML
En concentré*



*XML
Langage et applications*



*XML
Le guide de l'utilisateur*