Le XML
ou
e**X**tensible **M**arkup **L**anguage

QU'EST-CE QUE LE XML ?

XML est un langage informatique de **balisage générique, qui permet de définir un document XML où peut on stocker des données sous forme de Balises.**
décrire les données de manière aussi bien compréhensible par les hommes qui écrivent les documents XML que par les machines qui les exploitent.

Comparaison de XML & le HTML

En comparaison avec le HTML, qui permet de définir la structure d'un document HTML en utilisant des balises HTML5 et afficher des données, le XML permet de décrire la structure des données en utilisant des balises génériques (nos propres balises).

Document XML

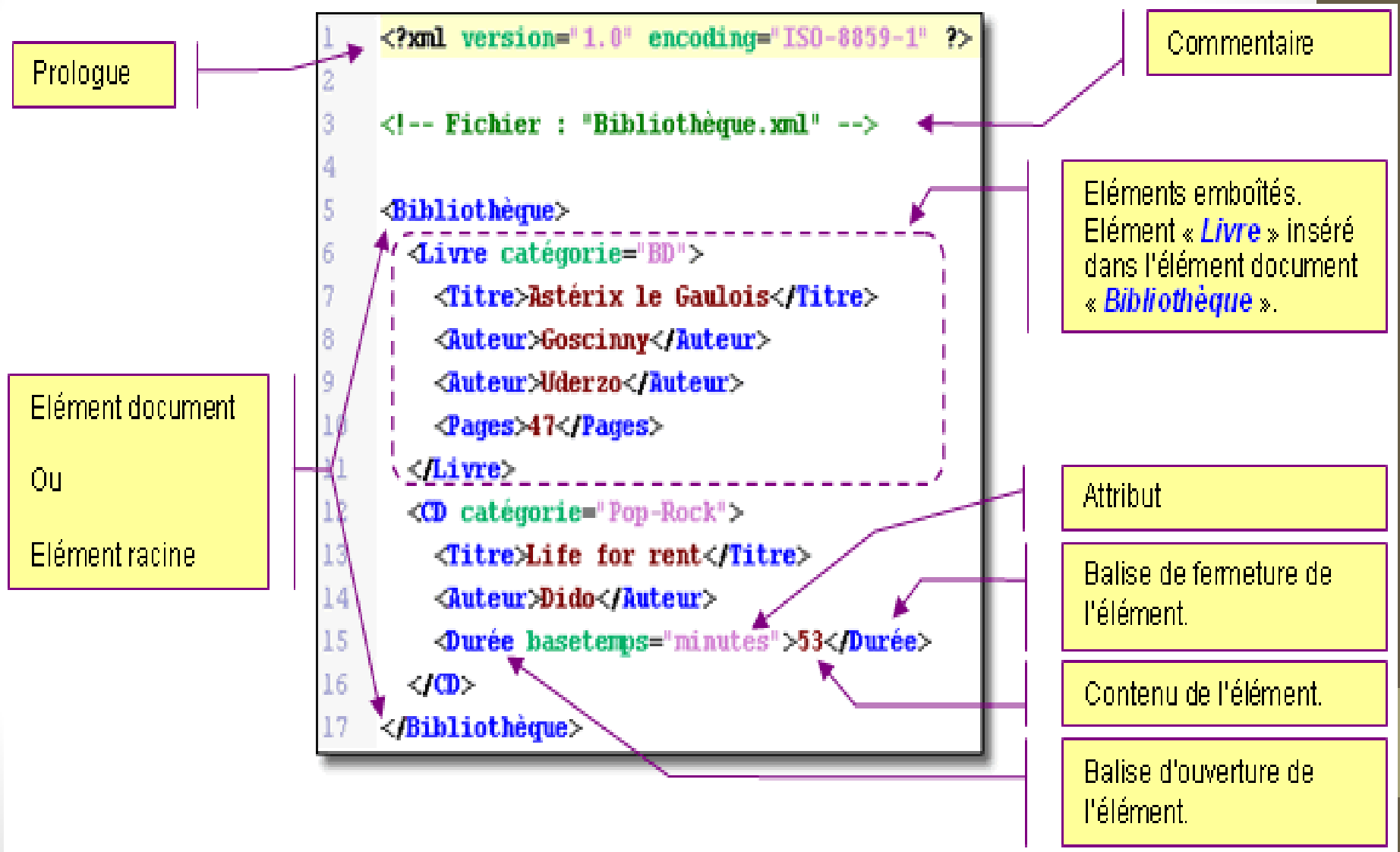
C'est Fichier d'extension XML qui est constitué par une arborescence de balises qui sont empilée les unes dans les autres.
Exemple:

```
<?xml version="1.0"  encoding="utf-8" ?>
<BIBLIOTHEQUE>
<LIVRE>
   <TITRE>titre livre 1</TITRE>
   <AUTEUR>auteur 1</AUTEUR>
   <EDITEUR>editeur 1</EDITEUR>
</LIVRE>
<LIVRE>
   <TITRE>titre livre 2</TITRE>
   <AUTEUR>auteur 2</AUTEUR>
   <EDITEUR>editeur 2</EDITEUR>
</LIVRE>
</BIBLIOTHEQUE>
```

Pourquoi XML?:

- Le XML a été créé pour faciliter les échanges de données entre les machines et les logiciels.
- Le XML est un langage qui s'écrit à l'aide de **balises**.
- Le XML est une recommandation du **W3C**, il s'agit donc d'une technologie avec des règles strictes à respecter.
- Le XML se veut compréhensible par tous : les hommes comme les machines.
- Le XML nous permet de créer notre propre vocabulaire grâce à un ensemble de règles et de balises personnalisables.

LES ÉLÉMENTS DE BASE D'UN DOCUMENT XML



Un document XML peut être découpé en 2 parties : le **prologue** et le **corps**:

Le prologue

<?xml version = "1.0" encoding="UTF-8" standalone="yes" ?>

Le prologue et contient, entre autre, la déclaration de la version XML employée pour la description des informations.

La version

Indiquer la version de XML que l'on utilise pour décrire nos données. Pour rappel, il existe actuellement 2 versions : 1.0 et 1.1.

Le jeu de caractères

Il s'agit du **jeu de caractères** utilisé dans le document XML

encoding="UTF-8".

encoding="ISO8859-1"

Les balises ou les éléments XML

Les balises par paires

<balise>Je suis le contenu de la balise</balise>

Une **balise par paires** peut également contenir une **autre balise**. On parle alors d'**arborescence**

```
<balise1>
        <balise2>10</balise2>
</balise1>
```

Enfin, une **balise par paires** peut contenir un **mélange de valeurs simples et de balises** comme en témoigne l'exemple suivant :

```
<balise1>
 Ceci est une chaîne de caractères
<balise2>10</balise2>
7.55
</balise1>
```

Les balises Vides

Une **balise unique** est en réalité une balise par paires qui n'a pas de contenu.

```
1    <élémentvide></élémentvide>
```

On peut écrire cet élément comme la balise **orphelines** qui suit:

```
1    <élémentvide />
```

Exemple:

```
1    <image source = "dessin.jpg" largeur = "250" hauteur = "150" />
```

Les règles de nommage des balises

Il y a cependant quelques règles de nommage à respecter pour les balises de votre langage balisé :

- Les noms peuvent contenir des lettres, des chiffres ou des caractères spéciaux.
- Les noms ne peuvent pas débuter par un nombre ou un caractère de ponctuation.
- Les noms ne peuvent pas commencer par les lettres XML (quelle que soit la casse).
- Les noms ne peuvent pas contenir d'espaces.
- On évitera les caractères **- , ; . <** et **>** qui peuvent être mal interprétés dans vos programmes.

Sensibilité à la casse

. La casse des noms doit être respectée.

Les attributs

Un attribut est une paire nom-valeur associée à la balise de début de l'élément.

```
1   <?xml version = "1.0" ?>
2
3   <!DOCTYPE bibliothèque SYSTEM "bibliothèque.dtd">
4
5   <bibliothèque>
6     <livre catégorie = "BD">
7       <titre>Gaston Lagafe</titre>
8       <auteur>Franquin</auteur>
9       <pages>61</pages>
10    </livre>
11    <livre catégorie = "Roman">
12          ...
13    </livre>
14  </bibliothèque>
```

Quelques règles

Tout comme pour les balises, quelques règles sont à respecter pour les attributs :

- Les règles de nommage sont les mêmes que pour les balises.
- La valeur d'un attribut doit impérativement être délimitée par des guillemets, simples ou doubles.
- Dans une balise, un attribut ne peut-être présent qu'une seule fois.

Les commentaires

<!-- Ceci est un commentaire ! -->

Un document bien formé

Cette notion décrit en réalité un document XML *conforme aux règles syntaxiques :*

- S'il s'agit d'un document utilisant la version 1.1 du XML, le prologue est bien renseigné.
- Le document XML ne possède qu'une seule balise racine.
- Le nom des balises et des attributs est conforme aux règles de nommage.
- Toutes les balises en paires sont correctement fermées.
- Toutes les valeurs des attributs sont entre guillemets simples ou doubles.
- Les balises de votre document XML ne se chevauchent pas, il existe une arborescence dans votre document.

Créer un nouveau document

- Démarrer le Rapid PHP ou autre éditeur.
- Ouvrir un nouveau fichier sous forme de fichier xml.
- Saisir votre document XML et l'enregistrer.

TP1:

Le but de ce TP est de créer un document XML structurant les données d'un répertoire.

Votre répertoire doit comprendre au moins 2 personnes. Pour chaque personne, on souhaite connaître les informations suivantes :

- Son sexe (homme ou femme).
- Son nom.
- Son prénom.
- Son adresse.
- Un ou plusieurs numéros de téléphone (téléphone portable, fixe, bureau, etc.).
- Une ou plusieurs adresses e-mail (adresse personnelle, professionnelle, etc.).

Rep:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<repertoire>
   <!-- John DOE -->
   <personne sexe="masculin">
      <nom>DOE</nom>
      <prenom>John</prenom>
      <adresse>
         <numero>7</numero>
         <voie type="impasse">impasse du chemin</voie>
         <codePostal>75015</codePostal>
         <ville>PARIS</ville>
         <pays>FRANCE</pays>
      </adresse>
      <telephones>
         <telephone type="fixe">01 02 03 04 05</telephone>
         <telephone type="portable">06 07 08 09 10</telephone>
      </telephones>
      <emails>
         <email type="personnel">john.doe@wanadoo.fr</email>
         <email type="professionnel">john.doe@societe.com</email>
      </emails>
   </personne>
```

```
<!-- Marie POPPINS -->
  <personne sexe="feminin">
    <nom>POPPINS</nom>
    <prenom>Marie</prenom>
    <adresse>
      <numero>28</numero>
      <voie type="avenue">avenue de la république</voie>
      <codePostal>13005</codePostal>
      <ville>MARSEILLE</ville>
      <pays>FRANCE</pays>
    </adresse>
    <telephones>
      <telephone type="professionnel">04 05 06 07 08</telephone>
    </telephones>
    <emails>
      <email type="professionnel">contact@poppins.fr</email>
    </emails>
  </personne>
</repertoire>
```

Exercice:

Distinguez les noms XML corrects des noms incorrects et corrigez les erreurs.

a. <Drivers_License_Number>98 NY 32</Drivers_License_Number>

b. <Driver's_License_Number>98 NY 32</Driver's_License_Number>

c. <month-day-year>7/23/2001</month-day-year>

d. <first name>Alan</first name>

e. <àçttûä>øåú</àçttûä>

f. <first_name>Alan</first_name>

g. <month/day/year>7/23/2001</month/day/year>

h. <_4-lane>I-610</_4-lane>

i. <téléphone>011 33 91 55 27 55 27</téléphone>

j. <4-lane>I-610</4-lane>

Réponse:

a. Correct
b. Incorrect (apostrophe)
c. Correct
d. Incorrect (présence d'un espace)
e. Correct
f. Correct
g. Incorrect (à cause des /)
h. Correct
i. Correct
j. Incorrect (un nom XML ne commence pas par un chiffre)

DTD

Document Type Description

Qu'est-ce que la définition d'un document XML ou le DTD?

Une **définition d'un document XML** est un ensemble de règles que l'on impose au document . Ces règles permettent de décrire la façon dont le document XML doit être construit. Elles peuvent être de natures différentes.

Par exemple, ces règles peuvent imposer la présence d'un attribut ou d'une balise, imposer l'ordre d'apparition des balises dans le document ou encore, imposer le type d'une donnée (nombre entier, chaîne de caractères, etc.).

En bref: DTD est un code qui permet de Contrôler le contenu d’un document XML, pour qu’il soit **VALIDE**

Où peut on écrire les DTD ?

Il existe 2 types de DTD : les **DTD externes** et les **DTD internes**.

DTD internes.

Une **DTD interne** est une DTD qui est écrite dans le même fichier que le document XML. Elle est généralement spécifique au document XML dans lequel elle est écrite.

```
<?xml version = "1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE  Racine [

<!– le Code DTD -- >

]>
```

Racine est la racine du document XML comme exemple:

```xml
<?xml version = "1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE boutique [
   <!-- code DTD --->
]>
<boutique>
  <telephone>
    <marque>Samsung</marque>
    <modele>Galaxy S3</modele>
  </telephone>

  <telephone>
    <marque>Apple</marque>
    <modele>iPhone 4</modele>
  </telephone>

  <telephone>
    <marque>Nokia</marque>
    <modele>Lumia 800</modele>
  </telephone>
</boutique>
```

Les DTD externes

Une **DTD externe** est une DTD qui est écrite dans un autre document que le document XML. Si elle est écrite dans un autre document, c'est que souvent, elle est commune à plusieurs documents XML qui l'exploitent.

Un fichier contenant uniquement une DTD porte l'extension **.dtd**.
Exemple:

```
<?xml version = "1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE boutique SYSTEM "dtd.dtd">
<boutique>
  <telephone>
    <marque>Samsung</marque>
    <modele>Galaxy S3</modele>
  </telephone>

  <telephone>
    <marque>Apple</marque>
    <modele>iPhone 4</modele>
  </telephone>

  <telephone>
    <marque>Nokia</marque>
    <modele>Lumia 800</modele>
  </telephone>
</boutique>
```

Création de la DTD

Les éléments:

Pour définir les règles portant sur les **balises**, on utilise le mot clef : ELEMENT.

<!ELEMENT balise (contenu)>

Exemple :

<nom>Mohamed</nom>

<!ELEMENT nom (contenu) >

Contenu : décrire ce que doit contenir la balise : est-ce une autre balise ou est-ce une valeur

Dans le cas où notre balise contient une **valeur simple**,(Une valeur simple désigne par exemple une chaîne de caractères, un entier, un nombre décimal, un caractère, etc.) on utilisera la mot clef #PCDATA

<!ELEMENT nom (#PCDATA) >

Pour un élément Vide en utilise : EMPTY

Exemple :

<!ELEMENT img EMPTY>

Cas d'une balise pouvant tout contenir : ANY

<!ELEMENT object ANY>

La séquence

Une **séquence** permet de décrire l'enchaînement imposé des balises.
Il suffit d'indiquer le nom des balises en les séparant par des *virgules*.

```
1    <!ELEMENT livre (titre, auteur, pages)>
```

Exemple:

```
<produit>
    <nom>XML par la pratique</nom>
    <prix>20</prix>
    Comment programmer des applications XML.
</produit>
```

Le DTD est :

```
<!ELEMENT produit (nom,prix,#PCDATA)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prix (#PCDATA)>
```

Prenons l'exemple suivant du DTD :

```
<!ELEMENT personne (nom, prenom, age)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT age (#PCDATA)>
```

Regardons alors la validité des documents XML qui suivent :

```
<personne>
   <nom>DOE</nom>
   <prenom>John</prenom>
   <age>24</age>
</personne>
```

Valide ou non valide??

```
<personne>
   <prenom>John</prenom>
   <nom>DOE</nom>
   <age>24</age>
</personne>
```

Valide ou non valide??

```
<personne>
   <prenom>John</prenom>
   <age>24</age>
</personne>
```

Valide ou non valide??

```
<personne>
   <nom>DOE</nom>
   <prenom>John</prenom>
   <age>24</age>
   <date>12/12/2012</date>
</personne>
```

Valide ou non valide??

La liste de choix

Une **liste de choix** permet de dire qu'une balise contient l'une des balises décrites. Il suffit d'indiquer le nom des balises en les séparant par une **barre verticale**.

```
<!ELEMENT personne (nom | prenom)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

```
<personne>
   <nom>DOE</nom>
</personne>
<personne>
   <prenom>John</prenom>
</personne>
<personne>
   <prenom>John</prenom>
   <nom>DOE</nom>
</personne>
```

La balise optionnelle

Une balise peut être **optionnelle**. Pour indiquer qu'une balise est optionnelle, on fait suivre son nom par un **point d'interrogation**.

```
<!ELEMENT personne (nom, prenom?)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

Cette définition impose que la balise <personne /> contienne obligatoirement la balise <nom /> puis éventuellement <prenom />.

```
<personne>
   <nom>DOE</nom>
</personne>
<personne>
   <nom>DOE</nom>
   <prenom>John</prenom>
</personne>
<personne>
   <prenom>John</prenom>
   <nom>DOE</nom>
</personne>
```

La balise répétée optionnelle: 0 ou plusieurs

Une balise peut être **répétée plusieurs fois** même si elle est optionnelle. Pour indiquer une telle balise, on fait suivre son nom par une **étoile**.

```
<!ELEMENT repertoire (personne*)>
<!ELEMENT personne (nom, prenom)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

```
<repertoire>
   <personne>
      <nom>DOE</nom>
      <prenom>John</prenom>
   </personne>
   <personne>
      <nom>POPPINS</nom>
      <prenom>Marie</prenom>
   </personne>
</repertoire>
```

La balise répétée:1 ou plusieurs

```
<!ELEMENT repertoire (personne+)>
<!ELEMENT personne (nom, prenom)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

? : autorise zéro ou un élément.
** : autorise zéro ou plusieurs éléments.*
+ : autorise un ou plusieurs éléments.

```
▼<infos_PC>
  ▼<modele>
      <nom>PC Home</nom>
      <prix_unitaire>5999</prix_unitaire>
      <fournisseur>Sulyo</fournisseur>
      <particularites>Base + lecteur DVD</particularites>
    </modele>
  ▼<modele>
      <nom>PC Multimedia</nom>
      <prix_unitaire>7999</prix_unitaire>
      <fournisseur>Fukoji</fournisseur>
      <particularites>PC HOME + ecran 19 + carte son</particularites>
    </modele>
</infos_PC>
```

```
<!DOCTYPE infos_PC [
 <!ELEMENT infos_PC (modele*)>
 <!ELEMENT modele (nom,prix_unitaire,fournisseur,particularites)>
 <!ELEMENT nom (#PCDATA)>
 <!ELEMENT prix_unitaire (#PCDATA)>
 <!ELEMENT fournisseur (#PCDATA)>
 <!ELEMENT particularites (#PCDATA)>
]>
```

DTD : LES ATTRIBUTS

Dans un DTD pour indiquer qu'une règle porte sur un **attribut**, on utilise le mot clef ATTLIST.

<!ATTLIST balise attribut type mode>

Exemple:

<personne sexe="masculin" />

Type: Est-ce une valeur bien précise ? Est-ce du texte ? Un identifiant ?

Cas d'un attribut ayant pour type la liste des valeurs possibles

<!ATTLIST personne sexe (masculin | féminin) mode>

Cas d'un attribut ayant pour valeur : un nombre, une lettre, une chaîne de caractères, etc.

<!ATTLIST personne sexe CDATA mode>

Cas d'un attribut ayant pour type un identifiant unique

<!ATTLIST personne num ID mode>

Cas d'un attribut ayant pour type une référence à un identifiant unique

```
<father id="PER-1" />
<child id="PER-2" father="PER-1" />

 <!ATTLIST father id ID mode >
 <!ATTLIST child id ID mode
             father IDREF mode
```

Mode : donner une information supplémentaire sur l'attribut comme par exemple une indication sur son obligation ou sa valeur.

attribut soit obligatoirement renseigné

<!ATTLIST personne sexe (masculin|féminin) #REQUIRED>

Cas d'un attribut optionnel

<!ATTLIST personne sexe CDATA #IMPLIED>

Cas d'une valeur par défaut

<!ATTLIST personne sexe CDATA "masculin">

Cas d'une constante

<!ATTLIST objet devise CDATA #FIXED "Euro">

Exemples:

```
<?xml version="1.0" encoding= "UTF-8" ?>
<!DOCTYPE personnes
 [
  <!ELEMENT personnes (personne+)>
  <!ELEMENT personne (nom,prenom)>
  <!ATTLIST personne sexe (masculin|feminin) #REQUIRED>
 ]>
 <personnes>
<personne sexe="masculin">
   <nom>PC Home</nom>
   <prenom>PC Home</prenom>
</personne>
<personne sexe="feminin">
  <nom>PC Home</nom>
   <prenom>PC Home</prenom>
</personne>
</personnes>
```

Exercices