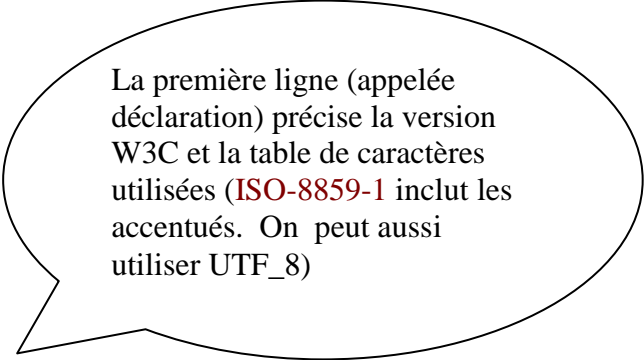


Pourquoi le XML ?

Tout comme le HTML, le XML a pour caractéristique d'être un langage « à balises ». Tous deux héritent des règles d'écriture du SGML qui constitue la norme d'écriture de ce type de langage. Au niveau syntaxique, la grande différence est que le XML va être « extensible » au niveau des balises utilisées ; autrement dit, c'est l'utilisateur qui définit les balises selon ses besoins. Par ailleurs, l'objectif du XML est tout autre que celui du HTML. Le but premier du XML est de fournir un outil d'**échange de données** universel entre divers programmes ou encore diverses plateformes ; En XML, aucune mise en forme des données n'est précisée.

Ainsi le fichier séquentiel suivant, dans lequel les données sont délimitées par des « , » gagnerait en lisibilité mais aussi en pouvoir d'évolutivité (changement de structure) s'il était écrit sous forme XML.

Cambier, Cédric, 064, 263542, 7160
Dupont, Alice, 065, 362456, 7000
Brancart, Alexandre, 067, 340552, 7060



La première ligne (appelée déclaration) précise la version W3C et la table de caractères utilisées (**ISO-8859-1** inclut les accentués. On peut aussi utiliser UTF_8)

Il devient en XML :

```
< ?xml version="1.0" encoding="ISO-8859-1" ?>
<Bottin>
<!--début du contenu du botin-->
  <Abonne>
    <Nom>Cambier</Nom>
    <Prenom>Cédric</Prenom>
    <Prefixe>064</Prefixe>
    <Tel>263542</Tel>
    <Adresse designation="rue" nom="du puits" numero="56"> </Adresse>
    <CodePostal>7160</CodePostal>
  </Abonne>

  <Abonne>
    <Nom>Dupont</Nom>
    <Prenom>Alice</Prenom>
    <Prefixe>065</Prefixe>
    <Tel>362456</Tel>
    <Adresse designation="avenue" nom="De Gaulle" numero="248"> </Adresse>
    <CodePostal>7000</CodePostal>
  </Abonne>

  <Abonne>
```

```

<Nom>Brancart</Nom>
<Prenom>Alexandre</Prenom>
<Prefixe>067</Prefixe>
<Tel>340552</Tel>
<Adresse designation="boulevard" nom="Sanders" numero="87"> </Adresse>
<CodePostal>7060</CodePostal>
</Abonne>
<!--fin du contenu du botin-->

</Bottin>

```

Principales contraintes syntaxiques du XML:

Fermeture des balises OBLIGATOIRES dans le but de rendre le document sans équivoque.

Un seul élément racine. (dans l'exemple c'est "Bottin")

Imbrication correcte des éléments "enfants" (ne pas inverser deux fermetures de balises)

Attributs obligatoirement entre "" ex: **encoding="ISO-8859-1" numero=" 87 "**

Respect de la casse (majuscule/minuscule) au niveau des balises.

Commentaires avec la balise <!-- -->

Entités standard: < signifie <
 > signifie >
 " signifie "
 ' signifie '
 & signifie &

Entités de caractères : &#codeAscii ; exemple é correspond au « é »

Concernant la première ligne du document .xml

Il s'agit des "Déclarations XML" indispensables dans un document XML afin que le "récepteur" et le "transmetteur" de données soient en accord.

< ?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>

Mot clé
sans
espace
entre ? et
xml

La version

Le type de codage : par
défaut="UTF-8"

Cfr www.unicode.org

Indique si le
document est
associé ou non à
un autre document
(cf. DTD par la
suite). Par défaut=
« no »

Balise (élément) ou attribut ?

La ligne `< Adresse designation="rue" nom="du puits" numero="56"> </Adresse>` peut aussi s'écrire: `<Adresse designation="rue" nom="du puits" numero="56"/>`

Elle aurait pu être écrite uniquement avec des éléments enfants de adresse, à savoir `<designation>` `<nom>` et `<numero>` plutôt qu'avec des attributs de adresse, c'est au choix du programmeur.

Disons qu'il serait intéressant de suivre les règles qui régissent l'approche objet, c'est-à-dire que lorsque cela correspond plutôt à une propriété qu'à une sous-classe, il est plus logique d'utiliser un attribut (avec valeur entre "").

Exercices à réaliser :

1) Encodage l'exemple ci-dessus (bottin) afin de vous familiariser aux contraintes d'écriture; si vous encodagez sans erreur..., faites-en expressément afin d'expérimenter les remarques faites précédemment. Testez l'ouverture du fichier dans un navigateur.

2) Proposez une autre solution plus "correcte" au niveau conception pour l'exemple du bottin et mettez-la en œuvre (approche objet- c'est-à-dire représentez les propriétés grâce à des attributs et non pas des nœuds)

3) Créez un document XML structurant les informations reprises dans l'annexe 1.0 concernant les stages.

4) Transformez le fichier Excel proposé en un document XML grâce à la programmation d'une macro VBA au sein du fichier Excel .