

Objectifs :

Ce TP est une initiation au langage XML. Il s'agira ici :

- de comprendre et de maîtriser les règles syntaxiques élémentaires du langage XML ;
- de mettre en œuvre un interpréteur XML (appelé "**parser**") pour créer un document XML (le parser XML sera celui fourni par la librairie Qt).

Pré-requis :

Initiation aux différents langages utilisés pour le développement de site Web, en particulier le langage HTML.

Conditions de réalisation :

Durée : 2 heures

Matériels et logiciels :

- un PC sous Windows avec l'outil de développement qtcreator, un navigateur et un accès internet.

Documentation :

On trouve beaucoup de documentations sur internet, certains de très bonnes qualités, je me contente de citer les deux références suivantes :

- le wiki sur le XML : http://fr.wikipedia.org/wiki/Extensible_Markup_Language ;
- <https://www.w3schools.com>

1. Présentation de XML

XML signifie **eXtensible Markup Language** et a été conçu pour **structurer** de l'information textuelle. Il fournit ainsi un format commun **d'échange** de données entre différentes applications.

Attention XML et HTML sont deux langages distincts : HTML a été conçu pour décrire de l'information en vue de son affichage dans un site web.

Mais comme HTML, XML est un langage à balises. Soit les deux exemples suivants :

<i>Exemple1.xml</i>	<i>Exemple2.xml</i>
<pre><?xml version="1.0" encoding="UTF-8"?> <!-- Document XML --> <Famille> <Père>Paul</Père> <Mère>Marie</Mère> <Enfant> <nom>Marine</nom> <sexe>Fille</sexe> <dateNaiss>22/10/2015</dateNaiss> </Enfant> <Enfant> <nom>Loïc</nom> <sexe>Garçon</sexe> <dateNaiss>12/05/2018</dateNaiss> </Enfant> </Famille></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <!-- Document XML --> <Famille> <Père>Paul</Père> <Mère>Marie</Mère> <Enfant nom="Marine" sexe="Fille" dateNaiss="22/10/2015"/> <Enfant nom="Loïc" sexe="Garçon" dateNaiss="12/05/2018"/> </Famille></pre>

De ces deux exemples on peut remarquer et déduire plusieurs règles :

1. Le nom des balises XML est totalement libre : cela permet à chacun de définir son propre langage pour décrire ses données (d'où le terme d'eXtensible).
2. La structure XML la plus simple est de la forme : **<Elément>Texte</Elément>**. Une balise XML s'appelle **Elément** et un Elément peut contenir du **Texte** ...
3. ...ou d'autres éléments. Il apparaît ainsi une arborescence où l'élément de plus haut niveau s'appelle **élément racine** (ici l'élément racine est l'élément *Famille*).
4. Un élément peut posséder des **attributs**. Il est possible alors de rencontrer les cas suivants :
 - a) L'élément *Personne* possède du **texte** :
`<Personne>Patrick DUPONT</Personne>` ;
 - b) L'élément *Personne* possède un **attribut** et du **texte** :
`<Personne sexe="M">Patrick DUPONT</Personne>` ;
 - c) L'élément *Personne* est vide mais possède un **attribut** :
`<Personne sexe="M" />` ;
 - d) L'élément *Personne* est vide et ne possède pas d'**attributs** :
`<Personne/>` ;
5. Un commentaire commence par `<!--` et se termine par `-->`.
6. Un document XML commence par une ligne de déclaration contenant la version et l'encodage, exemple :
`<?xml version="1.0" encoding="UTF-8"?>`.

Un document XML bien construit est simplement un document qui suit toutes les règles de notation et de structure du langage XML. Les programmes qui doivent traiter du XML devraient rejeter toute entrée de XML qui ne suit pas les règles de "bonne construction". Les plus importantes de ces règles sont les suivantes :

- Les balises sont sensibles aux majuscules et aux minuscules
- Aucune balise non terminée
- Les balises XML doivent être correctement imbriquées (pas de balises "croisées")
- Les valeurs des attributs doivent être données entre guillemets
- Les caractères de texte `<`, `>`, et `"` doivent toujours être représentés par des "entités de caractère"

Quelques informations sur XML :

XML et HTML ont une origine commune : SGML "*Standard Generalized Markup Language*" qui est un langage normalisé pour la génération de langage de balises. Les premières recherches sur les langages à balises ont été menées à la fin des années 60 (avec IBM qui créa le langage GML) pour aboutir à la norme SGML en 1986.

La structure d'un document XML peut-être décrite à l'aide d'un document DTD (Définition du Type de Document). On utilise pour cela la balise spéciale DOCTYPE.

XSL (eXtensible Stylesheet Language) est à XML ce que CSS (Cascading Style Sheet) est à HTML. XSL est un langage de style pour XML. Avec XSL on peut présenter ou afficher un document XML selon différentes présentations. Par exemple à l'aide d'un processeur XSLT (XSL Transformation) on peut convertir un document XML dans le format HTML, mais aussi dans le format PDF ou encore WML (pour les téléphones mobiles). Ainsi on sépare complètement le traitement du **contenu** (avec XML) et la **présentation** de ce contenu (avec XSL).

Exercice : Dans notepad++, créer le fichier **films.xml** à partir des informations contenues dans le fichier *ExerciceXML.txt*. Tester votre fichier XML avec un navigateur.

2. XML sous Qt

2.1 Créer un fichier XML

Vous allez créer un document XML en utilisant la librairie Qt. En effet on trouve dans Qt un ensemble de classes permettant de lire et de créer des documents XML. Ces classes XML mettent en œuvre ce que l'on appelle un interpréteur XML, appelé "**Parser**" (prononcer "parseur"). Un **parser** est un programme qui examine un document xml pour en extraire les données et la structure XML.

Bien que le but premier de Qt soit de fournir une bibliothèque graphique pour l'élaboration d'IHM de haute qualité, il est possible de créer des applications *Console*, ce qui nous permettra de tester les possibilités XML de QT sans s'occuper des aspects graphiques.

Vous trouverez dans les ressources du TP le fichier *FamilleXML.cpp* qui génère un fichier xml.

Exercices :

- 1) Examiner ce fichier *FamilleXML.cpp* et retrouver la construction du fichier *Exemple2.xml* cité ci-dessus.
Pour plus d'informations sur les classes utilisées, consulter l'aide de Qt.
- 2) Tester cet exemple. Pour cela créer un projet **Qt Console Application**.
Quel enfant manque-t-il ? Compléter l'exemple pour rajouter l'enfant manquant.
- 3) Dans un autre projet Qt, créer le fichier *Exemple1.xml*.

2.2 Lire un fichier XML

On veut lire dans un programme Qt le fichier **films.xml** créé précédemment.

Les classes Qt à utiliser pour le traitement XML sont *QDomDocument* et *QDomNode*.

On mémorisera les données des films dans les structures de données suivantes :

```
struct Realisateur{
    QString nom;
    QString prenom;
    QString annee_naissance;
};

struct Film{
    QString titre;
    QString annee;
    QString code_pays;
    QString genre;
    Realisateur realisateur;
    QString resume;
};

QVector<Film> Films;
```

Le programme se contentera d'afficher à l'écran les données des films se trouvant dans la variable *Films* de type *QVector<Film>*.

Exercice : Développer ce programme de lecture du fichier *films.xml*. On supposera le fichier *films.xml* bien formé.