

Le langage XML, les schémas XSD, Les transformations XSLT

Le langage XML (**EXtensible Markup Language**) a été défini par le par le W3C en février 1998, puis révisé en 2000.

Se référer aux cours relatifs au XML (DTD, CSS, XSLT) pour avoir tous les éléments de cours.



Olivier Mondet
<http://unidentified-one.net>

A. Pourquoi le langage XML ?



XML est un format d'échange de données structurées...

- utilisable sur n'importe que outil (un ordinateur, une imprimante, un téléphone portable...) ;
- utilisable avec de nombreux logiciels différents tels que des traitements de texte, des tableurs, des navigateurs (Word, Excel, la suite Open Office, Internet Explorer, Firefox...), des logiciels spécialisés conçus spécifiquement ;
- utilisable avec n'importe quelle plateforme libre ou commerciale (Linux, Windows, Mac OS, Symbian, Windows CE...) ;
- utilisable quelque soit le lieu et le jeu de caractères utilisés (car tous les types de caractères sont utilisables) ;
- évolutif (car c'est un standard qui s'impose à tous les logiciels conçus actuellement et ceux qui le seront demain).



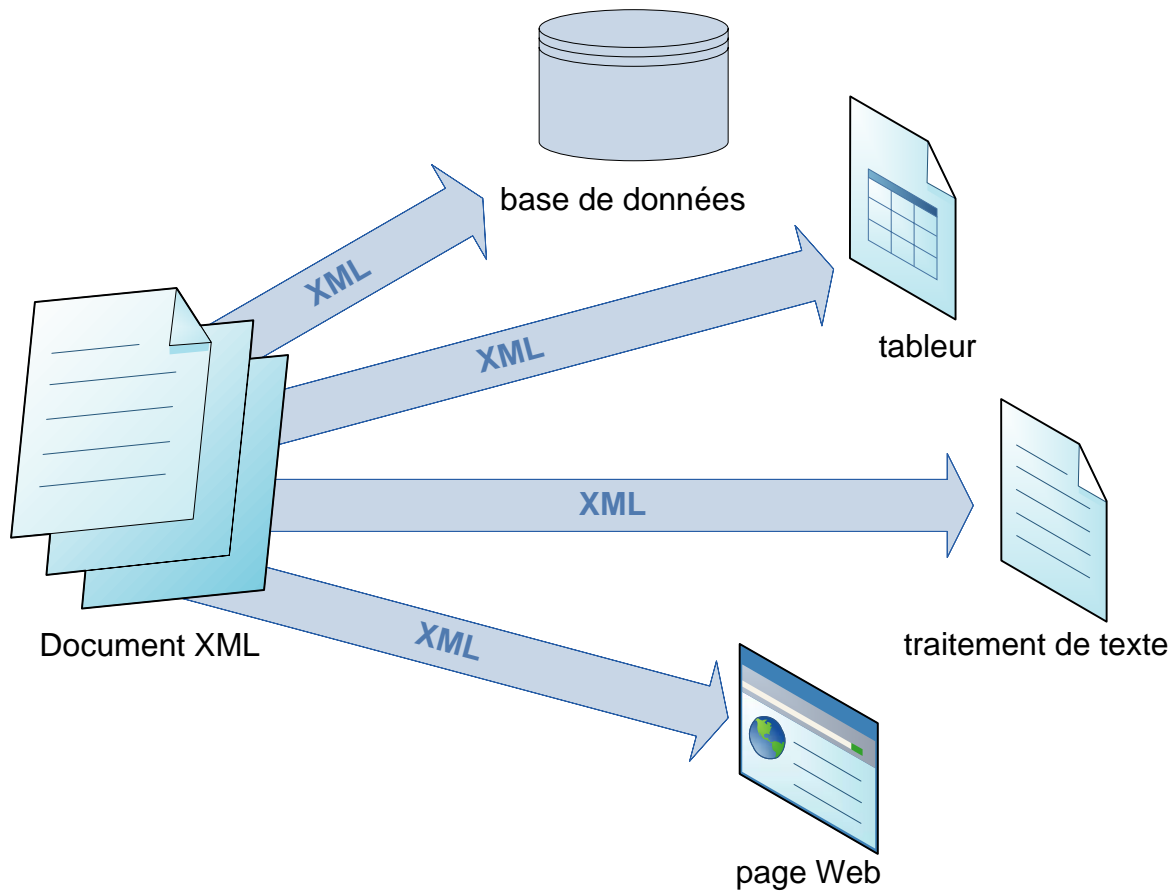
Concrètement, dans quel cas utilise t on un document XML ?

- pour faire des échanges de documents, d'informations (ex. des courriers, des données du personnel, des CV, des répertoires, des listes de critères d'embauche, des listes de candidatures reçues ou retenues...) ;
- pour avoir une multiplicité de logiciels sur lesquels peuvent être lus les documents (un traitement de texte, un tableur, un navigateur) ;
- pour avoir un document avec des règles de conception définies (les règles selon lesquelles on complète un document XML peuvent être définies pour que tous les intermédiaires suivent le même modèle grâce aux DTD ou aux schémas XSD) ;
- pour pouvoir archiver des informations qui pourront être lues à toutes époques, quelque soient les logiciels développés dans le futur (le XML est un standard libre, indépendant des médias et des environnements) ;
- pour ne pas avoir à payer un logiciel spécifique pour mes documents (de nombreux logiciels libres existent) ;
- chacun peut représenter différemment les informations contenues dans un document XML.

A terme le XML devrait être un format standard utilisé partout.



XML permet les échanges entre diverses sources d'information.



B. Concevoir un document XML

B.1. Structures XML

Un document XML doit commencer par une déclaration, qui indique l'utilisation du langage XML (version n°1.0) et le jeu de caractère latin (ISO-8859-1) que nous utilisons en typographie Française (caractères avec des accents par exemple).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
...
```

La description du contenu se fait grâce à des balises. Les balises encadrent le contenu de ce que l'on veut dire. Elles s'écrivent de cette manière : **<balise>contenu</balise>**

Ex1.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<annonce>
  Bonjour, je viens vous montrer un fichier écrit en XML.
</annonce>
```

Les balises encadrent ce que l'on appelle des **éléments**. Dans l'exemple n°1 l'élément est une annonce.

Les balises s'enchaînent en se contenant les unes par rapport aux autres. Elles ne doivent pas s'entrecouper. Le titre d'une balise doit être choisi en fonction de l'élément qu'elle décrit, sans accents, sans espaces, ni signes.

Ex2.

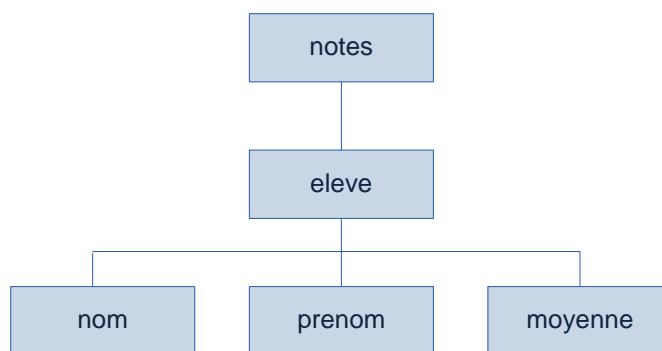
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<musique>
  <titre>Les marionnettes</titre>
  <titre>Petit papa Noël</titre>
</musique>
```

Avec les balises on peut donc composer des documents avec des contenus variés et structurés de façon arborescente. La structure d'un document XML doit obligatoirement débiter par un élément racine, comme ici :

Ex3a.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<notes>
  <eleve>
    <nom>ROBIN</nom>
    <prenom>Marc</prenom>
    <moyenne>12,5</moyenne>
  </eleve>
  <eleve>
    <nom>GRUNG</nom>
    <prenom>Stéphanie</prenom>
    <moyenne>11,3</moyenne>
  </eleve>
</notes>
```

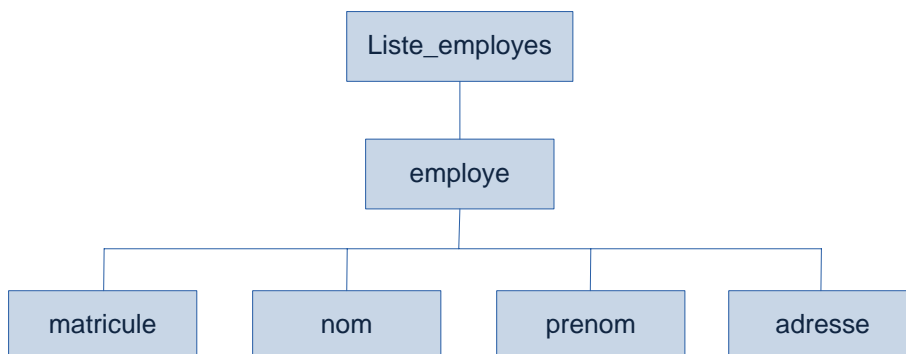
Ex3b. Structure des éléments



Ex4a.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<liste_employes>
  <employe>
    <matricule>23432</matricule>
    <nom>HUTIN</nom>
    <prenom>Sylvie</prenom>
    <adresse>16, rue des Passages - 91150 ETAMPES</adresse>
  </employe>
  <employe>
    <matricule>56435</matricule>
    <nom>JOLUN</nom>
    <prenom>Françoise</prenom>
    <adresse>112, avenue des Carreaux - 91580 ETRECHY</adresse>
  </employe>
</liste_employes>
```

Ex4a. Structure des éléments



Comme on peut le constater, la présentation du document est crénelée sur la gauche de façon à bien représenter l'arborescence. On appelle cela **indenter**.

On voit que certains éléments sont répétés : comme **eleve** dans l'exemple n°3 et **employe** dans l'exemple n°4. Les éléments **eleve** et **employe** décrivent des informations qui reviennent plusieurs fois comme les élèves d'une classe et les employés d'une organisation. D'autres éléments n'existent qu'une fois comme l'élément racine de chaque document.

Le XML donne une grande souplesse dans le choix de l'organisation des informations. Par exemple pour indiquer les différents numéros de téléphone que possède un individu :

Ex5a.

```
...
<portable>0610203040</portable>
<portable>0650607080</portable>
<domicile>0102030405</domicile>
...
```

Ex5b.

```
...
<telephone>
  <type>portable</type>
  <numero>0610203040</numero>
</telephone>
<telephone>
  <type>portable</type>
  <numero>0650607080</numero>
</telephone>
<telephone>
  <type>domicile</type>
  <numero>0102030405</numero>
</telephone>
```

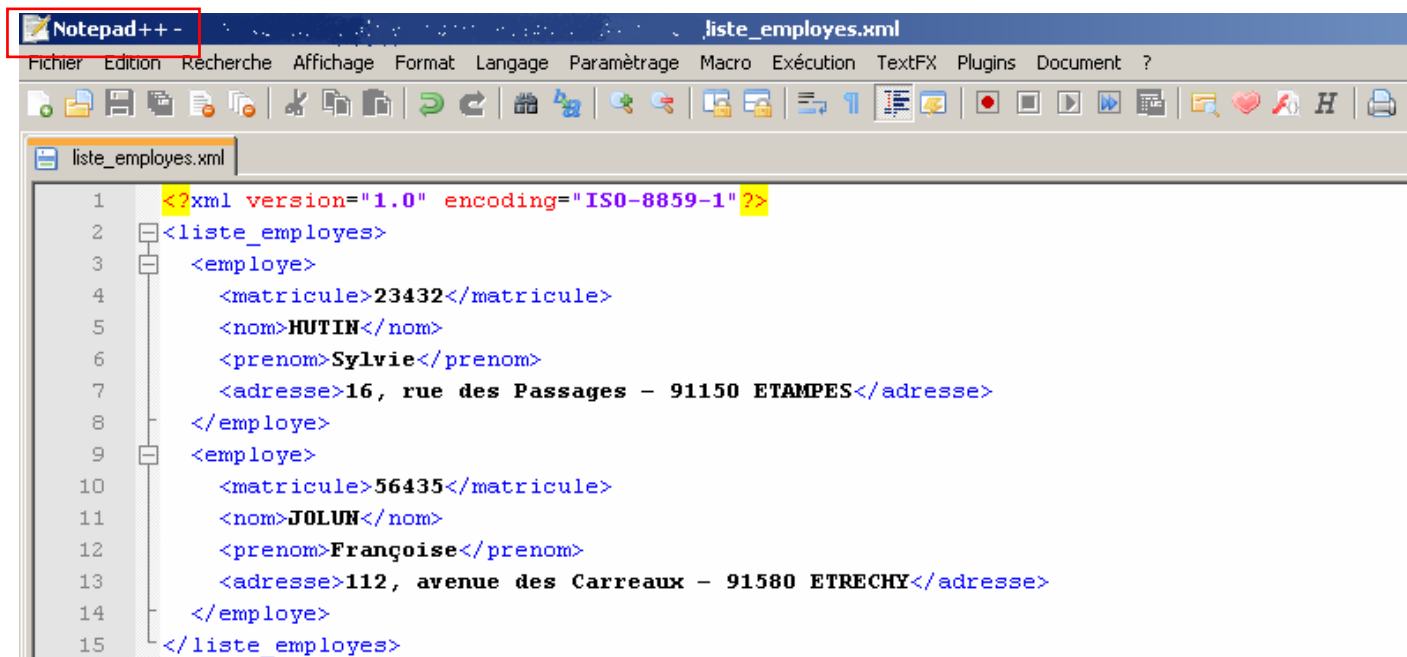
B.2. Comment créer un document XML ?

Il existe aujourd'hui de nombreux éditeurs qui permettent de créer des documents XML. Par exemple :

- **notepad++** qui est téléchargeable à <http://notepad-plus.sourceforge.net/>
- **XMLEditPro** qui est téléchargeable à <http://www.davewebsite.com/>
- **XMLNotePad** qui est téléchargeable à <http://www.devhood.com/>

Parmi les suites bureautiques : Office 2007 est compatible avec XML, Office 2003 Professionnel possède certaines fonctionnalités, OpenOffice (gratuit) est également compatible avec XML.

Tous les documents XML doivent avoir l'extension xml à la fin de leur nom. Par exemple : licenciements.xml, personnels.xml...



The screenshot shows the Notepad++ application window with the file 'liste_employes.xml' open. The code is as follows:

```
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <liste_employes>
3  <employe>
4    <matricule>23432</matricule>
5    <nom>HUTIN</nom>
6    <prenom>Sylvie</prenom>
7    <adresse>16, rue des Passages - 91150 ETAMPES</adresse>
8  </employe>
9  <employe>
10   <matricule>56435</matricule>
11   <nom>JOLUN</nom>
12   <prenom>Françoise</prenom>
13   <adresse>112, avenue des Carreaux - 91580 ETRECHY</adresse>
14 </employe>
15 </liste_employes>
```

B.3. Divers moyens de visualiser un document XML

Par définition XML, qui est format d'échange, n'est pas très esthétique puisqu'il ne représente que *l'information brute*. Il faudrait transformer le document XML pour lui donner une apparence plus conforme à ce qu'il devrait être. Tout dépend de la destination du document. Il sera représenté différemment si on l'utilise avec un tableur, un traitement de texte, une base de données, une page Web...

Sans transformation, on peut visualiser un document XML de diverses manières.

La première manière consiste à utiliser un éditeur comme notpad++ ou bien XMLEditPro (gratuits tous les deux).

La seconde manière consiste à utiliser les outils bureautiques ou bien le navigateur.

➔ L'exemple n° 4 dans Word (2003 Pro ou 2007).

The screenshot shows the Microsoft Word interface with the document 'liste_employes.xml' open. The main window displays the XML content with tags visible around the text. The 'Structure XML' task pane on the right provides a hierarchical view of the document's structure. The 'Afficher les balises XML dans le document' checkbox is checked. Below the tree view, there is a list of elements to apply to the selection, including 'adresse', 'employe', 'liste_employes', 'matricule', and 'nom'. The 'Afficher uniquement les éléments enfants de l'élément en cours' checkbox is unchecked and circled in red.

La structure est parfaitement reconnue, elle s'affiche avec une présentation proche du format d'origine. Word permet d'ajouter de nouveaux éléments, il faudra les ajouter en les sélectionnant dans la liste figurant en bas de la **Structure XML**. On pourra ainsi compléter le fichier facilement et graphiquement.



L'exemple n° 4 dans Excel (2003 Pro ou 2007).

The screenshot shows the Microsoft Excel 2003 interface with a spreadsheet titled 'Classeur4'. The spreadsheet has columns labeled A, B, C, D, and E. The data is as follows:

	A	B	C	D	E
1	matricule	nom	prenom	adresse	
2	23432	HUTIN	Sylvie	16, rue des Passages - 91150 ETAMPES	
3	56435	JOLUN	Françoise	112, avenue des Carreaux - 91580 ETRECHY	
4	*				

The 'Source XML' pane on the right shows the following structure:

- liste_employes_Mappage
 - liste_employes
 - employe
 - matricule
 - nom
 - prenom
 - adresse

The 'Liste' toolbar at the bottom of the spreadsheet area includes buttons for 'Liste', 'Afficher/Masquer la ligne Total', and other list-related functions.

La présentation avec Excel est structurée sous forme d'une feuille de calcul. Pour compléter le fichier XML il suffit de saisir une nouvelle ligne. Comme Word, Excel reconnaît l'existence d'un schéma XSD et peut également le générer.



L'exemple n° 4 dans Internet Explorer (6 ou 7).

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <liste_employes>
  - <employe>
    <matricule>23432</matricule>
    <nom>HUTIN</nom>
    <prenom>Sylvie</prenom>
    <adresse>16, rue des Passages - 91150 ETAMPES</adresse>
  </employe>
  - <employe>
    <matricule>56435</matricule>
    <nom>JOLUN</nom>
    <prenom>Françoise</prenom>
    <adresse>112, avenue des Carreaux - 91580 ETRECHY</adresse>
  </employe>
</liste_employes>

```

Le navigateur affiche la structure, il ne permet pas cependant de saisir de nouveaux éléments. Le navigateur (gratuit) Firefox permet d'afficher pareillement les structures XML.

C. Définir un document XML

C.1. Utilisation des schémas

Les documents XML peuvent être normés grâce à un schéma XSD (**XML Schema Definition**). Cela sert à donner des règles de structure pour un document XML. Lorsqu'un document XML est échangé entre deux partenaires, le fait de joindre un schéma permet à chacun de compléter le document de la même manière et avec la même rigueur. Sans cela l'échange pourrait être compromis ou n'aurait peut-être pas de sens.

Ex6a. Le document XML (liste_employes.xml)

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<liste_employes>
  <employe>
    <matricule>23432</matricule>
    <nom>HUTIN</nom>
    <prenom>Sylvie</prenom>
    <adresse>16, rue des Passages - 91150 ETAMPES</adresse>
  </employe>
  <employe>
    <matricule>56435</matricule>
    <nom>JOLUN</nom>
    <prenom>Françoise</prenom>
    <adresse>112, avenue des Carreaux - 91580 ETRECHY</adresse>
  </employe>
</liste_employes>

```

Le schéma va permettre, dans un premier temps, d'indiquer l'arborescence du document telle que nous l'avons représentée dans le point B.1.

Dans un second temps, le schéma permet d'indiquer le type des éléments du document :

string	Texte
data	Dates
integer	Nombre entier (1, 2, 3...)
decimal	Nombre réels (1.23, 0.17...)

Il existe de nombreux types décrits sur le site <http://www.w3.org/XML/Schema>

Dans un troisième temps, le schéma permet de préciser le nombre d'occurrences (nombre de fois ou sont répétés les éléments).

minOccurs=	"0" ou "1"
maxOccurs=	"1" ou "unbounded" (illimité)

Ex6b. Le schéma XSD (liste_employes.xsd)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="liste_employe">
    <xsd:complexType>
      <xsd:sequence maxOccurs="1" minOccurs="1">
        <xsd:element name="employe">
          <xsd:complexType>
            <xsd:sequence maxOccurs="unbounded" minOccurs="1">
              <xsd:element name="matricule" type="xsd:string"/>
              <xsd:element name="nom" type="xsd:string"/>
              <xsd:element name="prenom" type="xsd:string"/>
              <xsd:element name="adresse" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

C.2. Construction d'un schéma

Un schéma XSD doit commencer par une déclaration, qui indique l'utilisation du langage XML (comme les documents XML eux-mêmes), suivis d'une balise comportant l'indication d'un schéma global.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  ...
</xsd:schema>
```

C'est entre les deux balises indiquant l'utilisation d'un schéma que l'on déclare tous les éléments du document XML.

Le schéma est composé de balises, qui débutent toutes par **<xsd:** et se terminent toutes par **</xsd:**. Ces balises suivent les mêmes règles que toutes autres balises :

- toute balise ouverte doit être fermée ;
- pas d'enchevêtrement des balises ;
- pas de caractères accentués ou de signes dans l'écriture des balises.

Un document XML est composé d'éléments simples (destinés à recueillir des informations basiques comme un nom, une adresse, une date) et d'éléments complexes (composés d'autres éléments).

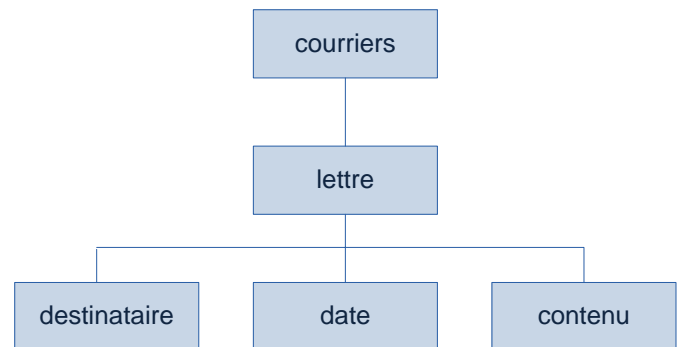


Déclaration d'un élément simple.

Un élément simple s'écrit comme suit : `<xsd:element name="MATRICULE" type="xsd:string"/>`
Où **name="MATRICULE"** correspond à la déclaration d'un élément nommé ELEMENT, et **type="xsd:integer"** correspond au type de contenu (entier) qui sera saisi entre les balises.

Ex7a.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<courriers>
  <lettre>
    <destinataire>Mme CORY</destinataire>
    <date>13/01/07</date>
    <contenu>Congé maternité</contenu>
  </lettre>
  <lettre>
    <destinataire>Mme TILLY</destinataire>
    <date>13/01/07</date>
    <contenu>Avancement</contenu>
  </lettre>
</courriers>
```



Dans cet exemple n°7a : **destinataire**, **date** et **contenu** sont des éléments simples qui seront déclarés comme suit :

Ex7b.

```
...
<xsd:element name="destinataire" type="xsd:string"/>
<xsd:element name="date" type="xsd:date"/>
<xsd:element name="contenu" type="xsd:string"/>
...
```

S'il se trouve qu'un élément doit être répété un certains nombre de fois, ou bien qu'il soit optionnel (il peut ne pas être présent ou bien une à plusieurs fois) on précise l'élément de la façon suivante :

`<xsd:element maxOccurs="4" minOccurs="1" name="TELEPHONE" type="xsd:integer"/>`

L'élément téléphone pourra être présent de 1 à 4 fois.

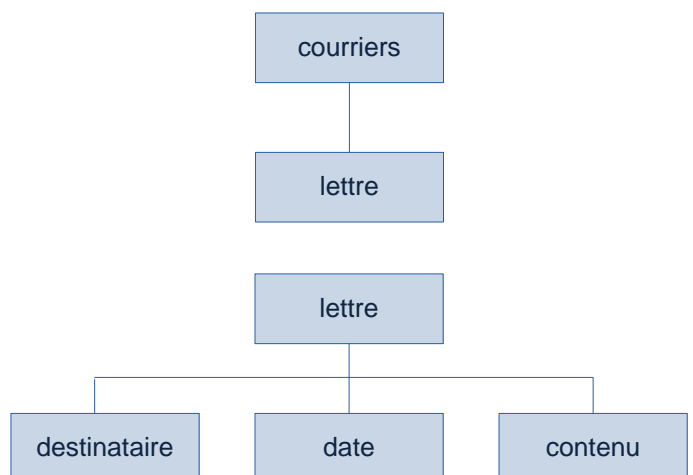


Déclaration d'un élément complexe.

Un élément complexe ne reçoit pas directement des informations, il contient soit d'autres éléments complexes, soit des éléments simples.

Dans l'exemple n°7a : **courrier** et **lettre** sont des éléments complexes :

- **courriers** contient un autre élément complexe -> **lettre**
- **lettre** contient des éléments simples -> **destinataire**, **date** et **contenu**



On aura tendance à déclarer sur plusieurs lignes un élément complexe, en indiquant :

- le nom de l'élément : `<xsd:element name="element">` ;
- le type complexe de l'élément : `<xsd:complexType>` ;
- le nombre de répétition : `<xsd:sequence maxOccurs="x" minOccurs="y">`.

Voici la déclaration de l'élément complexe **courriers** qui existe une seule fois :

Ex7c.

```
...
<xsd:element name="courriers">
<xsd:complexType>
<xsd:sequence maxOccurs="1" minOccurs="1">

</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Voici la déclaration de l'élément **courriers** et du sous élément **lettre** qui existe une à plusieurs fois :

Ex7d.

```
...
<xsd:element name="courriers">
<xsd:complexType>
<xsd:sequence maxOccurs="1" minOccurs="1">
  <xsd:element name="lettre">
    <xsd:complexType>
    <xsd:sequence maxOccurs="unbounded" minOccurs="1">
  ...
  </xsd:sequence>
  </xsd:complexType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```



Voici le schéma complet.

Ex7e.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="courriers">
    <xsd:complexType>
    <xsd:sequence maxOccurs="1" minOccurs="1">
      <xsd:element name="lettre">
        <xsd:complexType>
        <xsd:sequence maxOccurs="unbounded" minOccurs="1">
          <xsd:element name="destinataire" type="xsd:string"/>
          <xsd:element name="date" type="xsd:date"/>
          <xsd:element name="contenu" type="xsd:string"/>
        </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

C.3. Nécessité d'un schéma

Le schéma est optionnel. Il est rendu nécessaire dans des cas où l'on doit avoir une utilisation uniforme d'un document XML. Il existe des logiciels qui contrôlent si le document XML correspond bien à ce qui est écrit dans le schéma fourni avec (les logiciels de la suite Microsoft Office 2003 Professionnel le font).

De cette manière plus aucune erreur n'est possible, tous les interlocuteurs font la même chose.



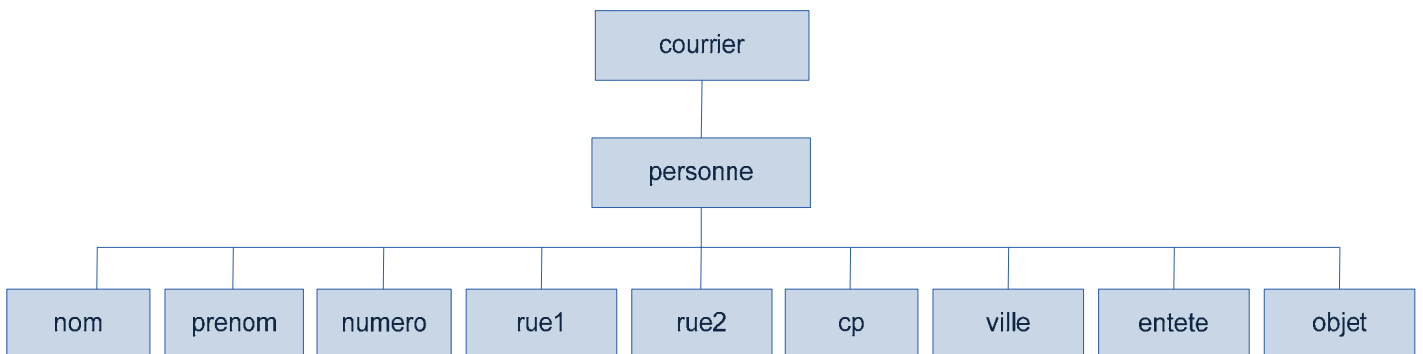
Courriers personnalisés à adresser (un extrait).

Ex8a. Le document XML (liste_grh.xml)

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

<courrier>
  <personne>
    <nom>Martin</nom>
    <prenom>Marie-Frédérique</prenom>
    <numero>123</numero>
    <rue1>Bâtiment B1</rue1>
    <rue2>allée des fleurs</rue2>
    <cp>91150</cp>
    <ville>Etampes</ville>
    <entete>Votre départ en retraite</entete>
    <objet>Très chère Madame, nous sommes heureux de vous annoncer votre départ
en retraite pour le 15/03/06.</objet>
  </personne>
  <personne>
    <nom>Moulin</nom>
    <prenom>Marie</prenom>
    <numero>13</numero>
    <rue1>allée des peupliers</rue1>
    <cp>91510</cp>
    <ville>Lardy</ville>
    <entete>Votre changement d'indice</entete>
    <objet>Très chère Madame, nous sommes heureux de vous annoncer votre
changement d'indice (n°454) à compter du 15/03/06.</objet>
  </personne>
</courrier>
```

Ex8b.



La structure, bien que plus étendue, reste classique. On notera simplement que **rue2** est un élément optionnel.

Ex8c. Le schéma XSD (liste_grh.xsd)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="COURRIER">
    <xs:complexType>
      <xs:sequence maxOccurs="1" minOccurs="1">
        <xs:element name="PERSONNE">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded" minOccurs="1">
              <xs:element maxOccurs="1" minOccurs="1" name="NOM" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="PRENOM" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="NUMERO" type="xs:integer"/>
              <xs:element maxOccurs="1" minOccurs="1" name="RUE1" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="0" name="RUE2" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="CP" type="xs:integer"/>
              <xs:element maxOccurs="1" minOccurs="1" name="VILLE" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="ENTETE" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="OBJET" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



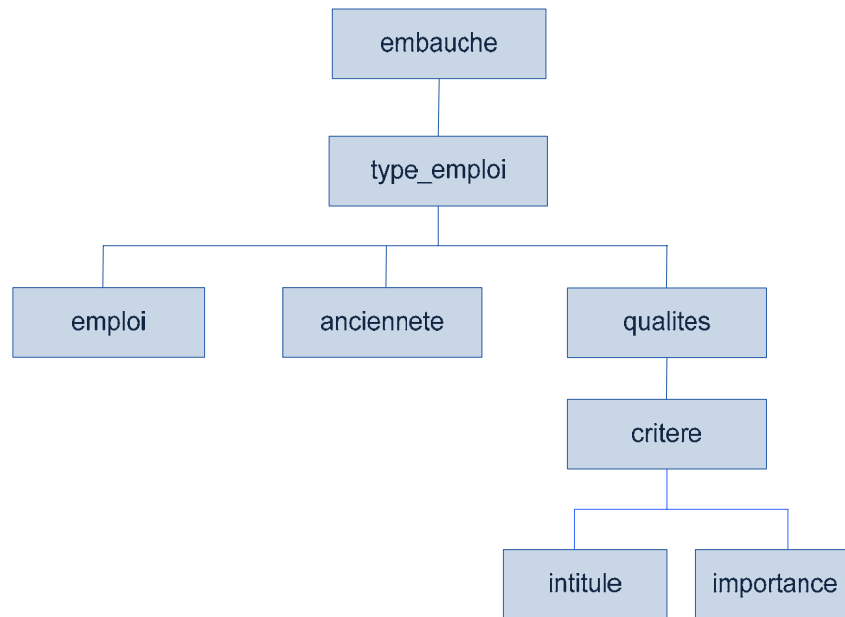
Liste des critères d'embauche (un extrait).

Ex9a. Le document XML (criteres_embauche.xml)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<embauche>
  <type_emploi>
    <emploi>assistant commercial</emploi>
    <anciennete>3 ans</anciennete>
    <qualites>
      <critere>
        <intitule>Mobilité</intitule>
        <importance>5</importance>
      </critere>
      <critere>
        <intitule>Anglais (lu)</intitule>
        <importance>3</importance>
      </critere>
    </qualites>
  </type_emploi>
  <type_emploi>
    <emploi>assistant comptable</emploi>
    <anciennete>5 ans</anciennete>
    <qualites>
      <critere>
        <intitule>Rigueur</intitule>
        <importance>5</importance>
      </critere>
      <critere>
        <intitule>Anglais (lu)</intitule>
        <importance>4</importance>
      </critere>
    </qualites>
  </type_emploi>
</embauche>
```

L'ancienneté représente le nombre d'années, importance représente l'importance d'un critère noté sur 5.

Ex9b.



Ex9c. Le schéma XSD (criteres_embauche.xsd)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="EMBAUCHE">
    <xs:complexType>
      <xs:sequence maxOccurs="1" minOccurs="1">
        <xs:element name="TYPE_EMPLOI">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded" minOccurs="1">
              <xs:element maxOccurs="1" minOccurs="1" name="EMPLOI" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="ANCIENNETE" type="xs:integer"/>
              <xs:element name="QUALITES"/>
            <xs:complexType>
              <xs:sequence maxOccurs="1" minOccurs="1">
                <xs:element name="CRITERE"/>
              <xs:complexType>
                <xs:sequence maxOccurs="unbounded" minOccurs="1">
                  <xs:element maxOccurs="1" minOccurs="1" name="INTITULE" type="xs:string"/>
                  <xs:element maxOccurs="1" minOccurs="1" name="IMPORTANCE" type="xs:integer"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

D. Transformation d'un document XML

Vous aurez pu vous en rendre compte, un document XML, même accompagné d'un schéma n'est pas satisfaisant au niveau de la présentation. Il y a deux raisons à cela :

- c'est un format d'échange donc on ne peut pas mettre en forme le document sachant que l'on ne sait pas ce qu'il deviendra (sera t-il lu sur un ordinateur ou un PDA ? Avec quel logiciel va-t-on l'utiliser ?...)
- seules les informations contenues comptent, la présentation n'est pas une information mais juste un moyen de représenter ces informations. On appelle cela **séparation du contenu et de la mise en forme**.

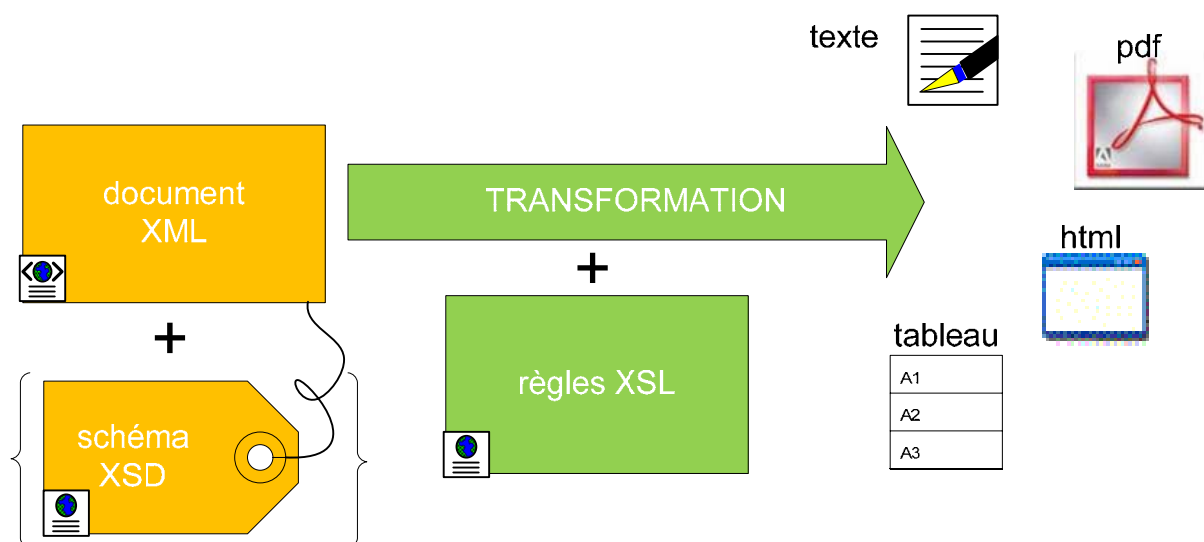
XML a la particularité d'être transformable. Transformable suivant la destination du document :

- Si on veut le lire avec un tableur, Open Office ou Excel à partir de la version 2003 Professionnel peuvent transformer les documents XML en feuille de tableur (avec mise en forme des cellules, etc.) ;
- Si on veut le lire avec un traitement de texte, Open Office ou Word à partir de la version 2003 Professionnel peuvent transformer les documents XML (avec mise en forme du texte, titrage, etc.) ;
- Si on veut le transformer en page Web, en document PDF, l'incorporer à une base de données, c'est possible également.

Sachez qu'à partir de la version 2007, tous les logiciels Microsoft Office travaillent nativement en XML, il en est de même pour Open Office depuis sa version 2. Ca veut dire qu'à tout document on peut ne garder que la partie XML, et se l'approprier avec n'importe quel logiciel, ou bien incorporer une présentation définie parallèlement.



On peut transformer un document XML.



D.1. Transformer un document XML en page HTML

Un document XML peut subir une transformation grâce à XSLT (**eXtended Stylesheet Language Transformations**).

Il faut créer un document XSL qui permettra de définir comment un document XML va être transformé. Nous nous appliquerons à ces transformations uniquement vers des pages HTML, mais sachez que l'on peut produire également des documents PDF, des documents pour les suites bureautiques Microsoft Office ou Open Office.

Nous verrons la transformation HTML vers le navigateur et vers le tableur Excel 2003 Professionnel.



Exemple - Gestion des heures des salariés.

Scénario : Soit une entreprise qui transmet les horaires des salariés via un document XML en vue de la préparation de la paie.

Ex10a. gestion_heures.xml

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

<?xml-stylesheet href="gestion_heures.xsl" type="text/xsl"?>

<horaires>
  ..<semaine>
    ....<libelle>33</libelle>
    ....<service>Atelier M23</service>
    .....<employe>
      .....<numero>657319</numero>
      .....<jour>
        .....<lib_jour>lundi</lib_jour>
        .....<debut>8h15</debut>
        .....<fin>17h36</fin>
        .....<pause>1h05</pause>
        .....</jour>
        .....<jour>
          .....<lib_jour>mardi</lib_jour>
          .....<debut>8h37</debut>
          .....<fin>17h30</fin>
          .....<pause>1h01</pause>
          .....</jour>
          .....<jour>
            .....<lib_jour>mercredi</lib_jour>
            .....<debut>8h08</debut>
            .....<fin>17h12</fin>
            .....<pause>1h00</pause>
            .....</jour>
            .....<jour>
              .....<lib_jour>jeudi</lib_jour>
              .....<debut>8h33</debut>
              .....<fin>17h01</fin>
              .....<pause>1h12</pause>
              .....</jour>
              .....<jour>
                .....<lib_jour>vendredi</lib_jour>
                .....<debut>8h10</debut>
                .....<fin>16h03</fin>
                .....<pause>0h55</pause>
                .....</jour>
            .....</employe>
            .....<employe>
              .....<numero>740051</numero>
              .....<jour>
                .....<lib_jour>lundi</lib_jour>
                .....<debut>8h34</debut>
                .....<fin>17h40</fin>
                .....<pause>0h55</pause>
                .....</jour>
                .....<jour>
                  .....<lib_jour>mardi</lib_jour>
                  .....<debut>8h17</debut>
                  .....<fin>17h28</fin>
                  .....<pause>1h01</pause>
                  .....</jour>
                  .....<jour>
                    .....<lib_jour>mercredi</lib_jour>
                    .....<debut>8h45</debut>
```

```

.....<fin>17h56</fin>
.....<pause>0h53</pause>
.....</jour>
.....<jour>
.....<lib_jour>jeudi</lib_jour>
.....<debut>8h22</debut>
.....<fin>17h45</fin>
.....<pause>0h45</pause>
.....</jour>
.....<jour>
.....<lib_jour>vendredi</lib_jour>
.....<debut>8h39</debut>
.....<fin>16h11</fin>
.....<pause>0h59</pause>
.....</jour>
....</employe>
..</semaine>
</horaires>

```

Pour sortir un état des horaires et faire des contrôles, on applique des règles XSL en vue de transformer le document XML en page Web (construite avec du HTML).

Ex10b. gestion_heures.xsl

```

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

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="iso-8859-1" doctype-public="-//W3C//DTD XHTML 1.0
Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"/>
<xsl:template match="/">
<html>
..<head>
..<title>Etat des heures</title>
..</head>
..<body>
..<h1>Etat des heures</h1>
..<xsl:for-each select="horaires/semaine">
....<h2>Semaine n° <xsl:value-of select="libelle"/></h2>
....<h2>Service n° <xsl:value-of select="service"/></h2>
....
....<xsl:for-each select="employe">
.....<h3>Salarié : <xsl:value-of select="numero"/></h3>
.....<table border="3">
.....<tr>
.....<td>jour</td>
.....<td>heure début</td>
.....<td>heure fin</td>
.....<td>temps pause</td>
.....</tr>
.....<xsl:for-each select="jour">
.....<tr>
.....<td><xsl:value-of select="lib_jour"/></td>
.....<td><xsl:value-of select="debut"/></td>
.....<td><xsl:value-of select="fin"/></td>
.....<td><xsl:value-of select="pause"/></td>
.....</tr>
.....</xsl:for-each>
.....</table>
....</xsl:for-each>
..</xsl:for-each>
..</body>
</html>
</xsl:template>
</xsl:stylesheet>

```


Etat des heures

Semaine n° 33

Service n° Atelier M23

Salarié : 657319

jour	heure début	heure fin	temps pause
lundi	8h15	17h36	1h05
mardi	8h37	17h30	1h01
mercredi	8h08	17h12	1h00
jeudi	8h33	17h01	1h12
vendredi	8h10	16h03	0h55

Salarié : 740051

jour	heure début	heure fin	temps pause
lundi	8h34	17h40	0h55
mardi	8h17	17h28	1h01
mercredi	8h45	17h56	0h53
jeudi	8h22	17h45	0h45
vendredi	8h39	16h11	0h59

Terminé

Poste de travail

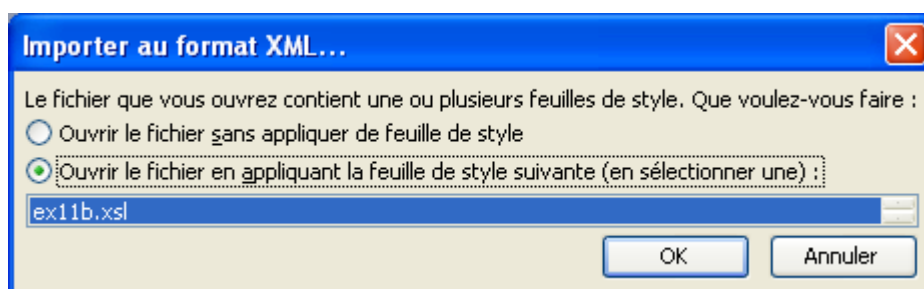
Ex10d. code HTML généré (extrait)

```

<html>
<head>
<title>Etat des heures</title>
</head>
<body>
<h1>Etat des heures</h1>
<h2>Semaine n° 33</h2>
<h2>Service n° Atelier M23</h2>
<h3>Salarié : 657319</h3>
<table border="3">
<tr>
<td>jour</td>
<td>heure début</td>
<td>heure fin</td>
<td>temps pause</td>
</tr>
<tr>
<td>lundi</td>
<td>8h15</td>
<td>17h36</td>
<td>1h05</td>
</tr>
<tr>
...
</table>
</body>
</html>

```

Avec Excel 2003 Professionnel, vous pouvez ouvrir votre document XML. Dès l'ouverture Excel propose d'ouvrir le document de façon à voir les données seules (on pourra enrichir et modifier le document) ou d'opérer une transformation avec le document XSL :



Une fois choisie la feuille de style, le tableur affiche le résultat suivant :

Etat des heures

Semaine n° 33

Service n° Atelier M23

Salarié : 657319

jour	heure début	heure fin	temps pause
lundi	8h15	17h36	1h05
mardi	8h37	17h30	1h01
mercredi	8h08	17h12	1h00
jeudi	8h33	17h01	1h12
vendredi	8h10	16h03	0h55

Salarié : 740051

jour	heure début	heure fin	temps pause
lundi	8h34	17h40	0h55
mardi	8h17	17h28	1h01
mercredi	8h45	17h56	0h53
jeudi	8h22	17h45	0h45
vendredi	8h39	16h11	0h59



Mécanisme de la transformation.

Le navigateur Internet Explorer, comme le logiciel Excel ont opéré la transformation du document **gestion_heures.xml** en page HTML grâce aux instructions contenues dans **gestion_heures.xsl**.



On pourra également utiliser XMLEditPro pour réaliser cette transformation.



D.2. Comment concevoir des règles de transformation ?

La structure de la page Web est supposée être connue, voici sa structure de base :

```
<head>
  <title>Titre de votre page</title>
</head>
<body>
  ...
</body>
</html>
```

Il faudra jouter les déclarations nécessaires suivant que l'on fera un document HTML ou XHTML.

Les règles XSL sont déclarées dans un document XML.

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

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="iso-8859-1" doctype-public="-//W3C//DTD XHTML 1.0
Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"/>
<xsl:template match="/">
  ...
</xsl:template>
</xsl:stylesheet>
```



Les instructions.

<code><xsl:template match="element"></code> ... <code></xsl:template></code>	Est évoqué dès que la racine "/" du fichier XML à transformer est rencontrée. L'attribut mach sert à indiquer un élément particulier
<code><xsl:for-each select="element"></code> ... <code></xsl:for-each></code>	Indique une action qui sera répétée pour chaque élément.
<code><xsl:value-of select="element"/></code>	Retourne la valeur de l'élément sélectionné.
<code><xsl:template name="nom_t"></code> ... <code></xsl:template></code>	Permet de créer une fonction "nom_t".
<code><xsl:call-template name="nom_t"/></code>	Permet d'appeler une fonction.



Exemple - Fichier du personnel (sous forme de tableau).

Scénario : Une entreprise gère son fichier du personnel avec un document XML.

Ex11a. fichier_personnel.xml

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

<?xml-stylesheet href="fichier_personnel.xsl" type="text/xsl"?>
<personnel>
  ..<personne>
    ....<nom>Martin</nom>
    ....<prenom>Marie-Fred</prenom>
    ....<coordonnees>
      .....<numero>123</numero>
      .....<rue1>allée des fleurs</rue1>
      .....<telephone>
        .....<type>bureau</type>
        .....<numero>0123456789</numero>
        .....</telephone>
        .....<telephone>
          .....<type>domicile</type>
          .....<numero>0198765432</numero>
          .....</telephone>
        ....</coordonnees>
      ..</personne>
    ..<personne>
      ....<nom>Moulin</nom>
      ....<prenom>Jean</prenom>
      ....<coordonnees>
        .....<numero>78bis</numero>
        .....<rue1>allée des Chevaux</rue1>
        .....<telephone>
          .....<type>bureau</type>
          .....<numero>0223456789</numero>
          .....</telephone>
          .....<telephone>
            .....<type>domicile</type>
            .....<numero>0298765432</numero>
            .....</telephone>
          ....</coordonnees>
        ..</personne>
      ..</personnel>
```

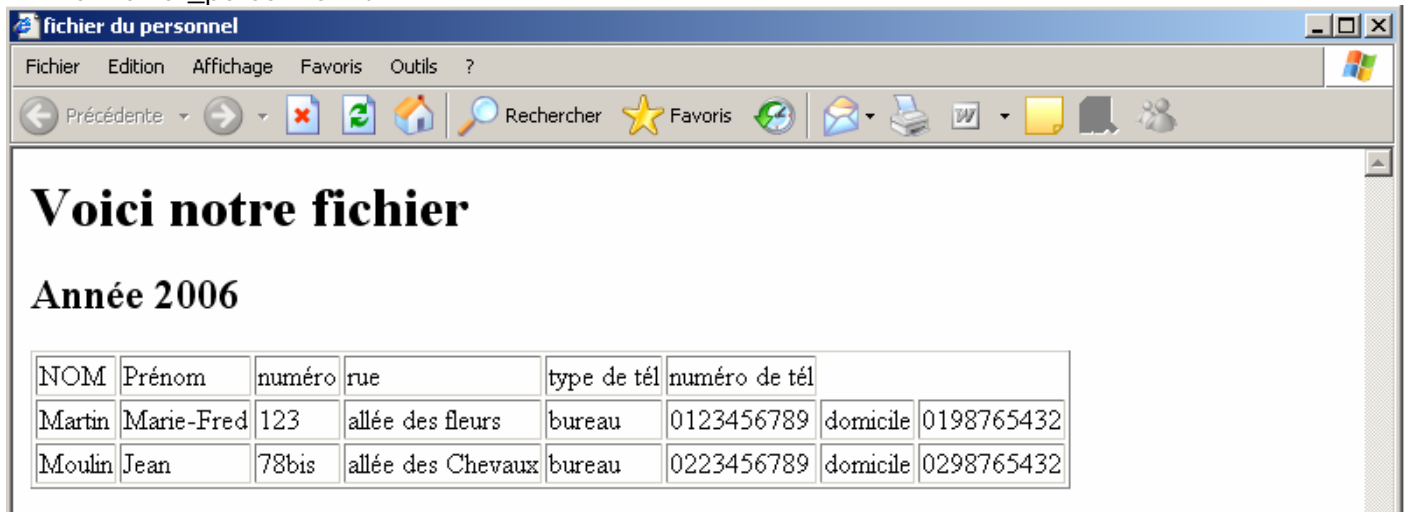
Ex11b. fichier_personnel.xsl

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

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="iso-8859-1" doctype-public="-//W3C//DTD XHTML 1.0
Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"/>
<xsl:template match="/">
<html>
..<head>
..<title>fichier du personnel</title>
..</head>
..<body>
..<h1>Voici notre fichier</h1>
..<h2>Année 2006</h2>
..<table border="1">
....<tr>
.....<td>NOM</td>
.....<td>Prénom</td>
.....<td>numéro</td>
.....<td>rue</td>
.....<td>type de tél</td>
.....<td>numéro de tél</td>
....</tr>
....<xsl:for-each select="personnel/personne">
....<tr>
.....<td><xsl:value-of select="nom"/></td>
.....<td><xsl:value-of select="prenom"/></td>
.....<td><xsl:value-of select="coordonnees/numero"/></td>
.....<td><xsl:value-of select="coordonnees/rue1"/></td>
.....<xsl:for-each select="coordonnees/telephone">
.....<td><xsl:value-of select="type"/></td>
.....<td><xsl:value-of select="numero"/></td>
.....</xsl:for-each>
....</tr>
....</xsl:for-each>
..</table>
..</body>
</html>

</xsl:template>
</xsl:stylesheet>
```

Ex11c. fichier_personnel.html



La version d'Excel 2003 :

The screenshot shows the Microsoft Excel 2003 interface. The title bar reads 'Microsoft Excel - ex11a.xml [Lecture seule]'. The menu bar includes 'Fichier', 'Edition', 'Affichage', 'Insertion', 'Format', 'Outils', 'Données', and 'Fenêtre'. The toolbar contains various icons for file operations and editing. The formula bar shows 'K13' and a formula icon. The spreadsheet grid has columns A through I. Row 1 contains the text 'Voici notre fichier'. Row 2 is empty. Row 3 contains the text 'Année 2006'. Row 4 is empty. Row 5 is the header of a table with columns: 'NOM', 'Prénom', 'numéro', 'rue', 'type de tél', 'numéro de tél', and 'domicile'. Row 6 contains the data: 'Martin', 'Marie-Fred', '123', 'allée des fleurs', 'bureau', '123456789', 'domicile', '198765432'. Row 7 contains the data: 'Moulin', 'Jean', '78bis', 'allée des Chevaux', 'bureau', '223456789', 'domicile', '298765432'. Row 8 is empty.



Exemple - Critères d'embauche (sous forme de liste à puces).

Scénario : Une échange avec ses filiales les critères d'embauche du groupe pour uniformiser le recrutement.

Ex12a. critere_emb.xml

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

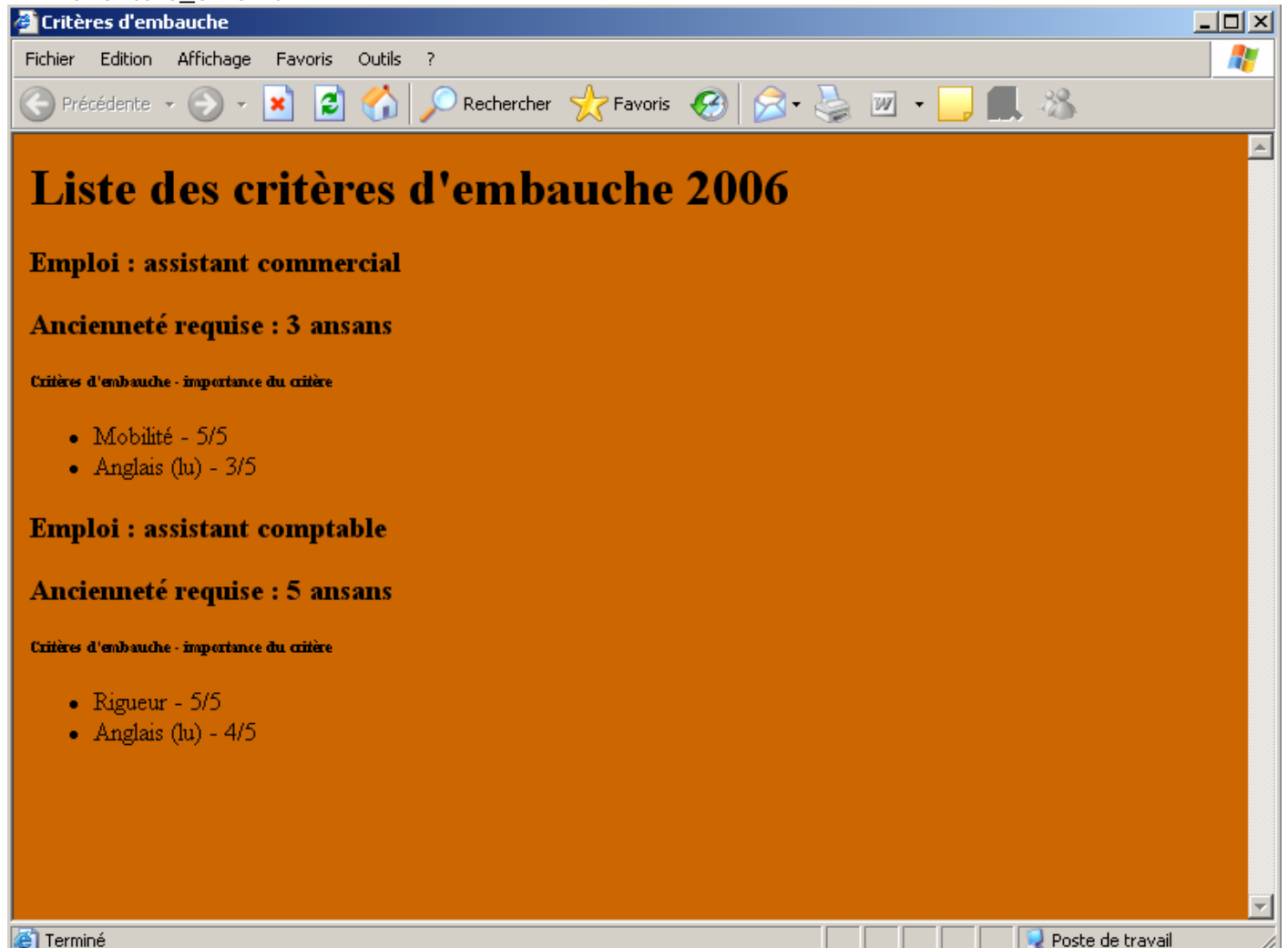
<?xml-stylesheet href="critere_emb.xsl" type="text/xsl"?>
<embauche>
  ..<type_emploi>
  ....<emploi>assistant commercial</emploi>
  ....<anciennete>3 ans</anciennete>
  ....<qualite>
  .....<critere>
  .....<intitule>Mobilité</intitule>
  .....<importance>5/5</importance>
  .....</critere>
  .....<critere>
  .....<intitule>Anglais (lu)</intitule>
  .....<importance>3/5</importance>
  .....</critere>
  ....</qualite>
  ..</type_emploi>
  ..<type_emploi>
  ....<emploi>assistant comptable</emploi>
  ....<anciennete>5 ans</anciennete>
  ....<qualite>
  .....<critere>
  .....<intitule>Rigueur</intitule>
  .....<importance>5/5</importance>
  .....</critere>
  .....<critere>
  .....<intitule>Anglais (lu)</intitule>
  .....<importance>4/5</importance>
  .....</critere>
  ....</qualite>
  ..</type_emploi>
</embauche>
```

Ex12b. critere_emb.xml

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

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="iso-8859-1" doctype-public="-//W3C//DTD XHTML 1.0
Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"/>
<xsl:template match="/">
..<html>
....<head>
.....<title>Critères d'embauche</title>
....</head>
....<body bgcolor="#CC6600">
.....<h1>Liste des critères d'embauche 2006</h1>
.....<xsl:for-each select="embauche/type_emploi">
.....<h3>Emploi : <xsl:value-of select="emploi"/></h3>
.....<h3>Ancienneté requise : <xsl:value-of select="anciennete"/>ans</h3>
.....<h6>Critères d'embauche - importance du critère</h6>
.....<ul>
.....<xsl:for-each select="qualite/critere">
.....<li><xsl:value-of select="intitule"/> - <xsl:value-of
select="importance"/></li>
.....</xsl:for-each>
.....</ul>
.....</xsl:for-each>
....</body>
..</html>
</xsl:template>
</xsl:stylesheet>
```

Ex12b. critere_emb.html



La version d'Excel 2003 :

Microsoft Excel - ex1 2a.xml [Lecture seule]

Fichier Edition Affichage Insertion Format Outils Données Fenêtre ?

Arial 10 G I S

K7 fx

1 **Liste des critères d'embauche 2006**

2

3 **Emploi : assistant commercial**

4

5 **Ancienneté requise : 3 ansans**

6

7 **Critères d'embauche - importance du critère**

8

9 Mobilité - 5/5

10 Anglais (lu) - 3/5

11

12 **Emploi : assistant comptable**

13

14 **Ancienneté requise : 5 ansans**

15

16 **Critères d'embauche - importance du critère**

17

18 Rigueur - 5/5

19 Anglais (lu) - 4/5

20

21

22

23

24

25

26

27

28

29

ex12a