

XML, DTD & XSLT

■ ■ ■ Utilisation de XML

1 – Un revendeur de pièces automobiles pour véhicule de collection, qui est spécialisé et limité à la « Coccinelle » de Volkswagen, la 2CV de Citroën et la Mini de BMC, veut mettre la liste des pièces dont il dispose sur un site internet.

- a. Donnez un DTD pour un fichier contenant des pièces individuelles, en essayant d'éviter au maximum les erreurs de saisie qui pourrait conduire à des informations « identiques » mais écrites avec des variations.

Le fichier « ListePieces.dtd »:

```
<!ELEMENT ListePieces (Piece+)>
<!ELEMENT Piece (Description,Prix)>
<!ELEMENT Prix (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ATTLIST Piece
  reference ID #REQUIRED
  modele (2CV|Mini|Coccinelle) "2CV"
  disponibilite (unique|
  commandable_chez_constructeur) "unique"
  etat (neuf|stock_inventu|occasion) "neuf">
```

Le fichier « mon_stock.xml »:

```
<?xml version="1.0" ?>
<!DOCTYPE ListePieces SYSTEM "ListePieces.dtd">
<ListePieces>
<Piece reference="P834" modele="2CV" disponibilite="unique" etat="neuf">
<Description>Pot d'échappement
</Description>
<Prix>150€</Prix>
</Piece>
<Piece reference="T563" modele="Mini"
disponibilite="unique" etat="neuf">
<Description>sigle</Description>
<Prix>50€</Prix>
</Piece>
<Piece reference="M546" modele="2CV" disponibilite="unique" etat="occa
sion">
<Description>volant</Description>
<Prix>100€</Prix>
</Piece>
</ListePieces>
```

On obtient aucune erreur en sortie de la validation par la commande « xmllint »:

```
xterm
xmllint mon_stock.xml --dtdvalid ListePieces.dtd -noout
```

b. Afin d'afficher le document XML sur le site web de la boutique

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/ListePieces">
<html><head><title>Boutique Voitures Collection</title></head>
<body>
<h1>2CV</h1>
<xsl:value-of select="count(Piece[@modele='2CV'])"/> pièces
<table><tr><td>Description</td><td>État</td><td>Prix</td></tr>
<xsl:for-each select="Piece[@modele='2CV']">
<xsl:sort select="@reference"/>
<tr><td><xsl:value-of select="Description"/></td>
<td><xsl:value-of select="@etat"/></td>
<td><xsl:value-of select="Prix"/></td></tr>
</xsl:for-each>
</table>
<h1>Mini</h1>
<xsl:value-of select="count(Piece[@modele='Mini'])"/> pièces
<table><tr><td>Description</td><td>État</td><td>Prix</td></tr>
<xsl:for-each select="Piece[@modele='Mini']">
<xsl:sort select="@reference"/>
<tr><td><xsl:value-of select="Description"/></td>
<td><xsl:value-of select="@etat"/></td>
<td><xsl:value-of select="Prix"/></td></tr>
</xsl:for-each>
</table>
</body></html>
</xsl:template>
</xsl:stylesheet>
```

Ce qui donne le résultat suivant avec la commande « xsltproc » :

```
xterm
pef@samus:~/Compilation2/TD3$ xsltproc web.xslt mon_stock.xml
```

c. Le revendeur voudrait maintenant pouvoir gérer plusieurs pièces individuelles comme un lot

```
<!ELEMENT ListePieces (Piece|Lot)+>
<!ELEMENT Piece (Description,Prix)>
<!ELEMENT Prix (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!--ATTLIST Piece
reference ID #REQUIRED
modele (2CV|Mini|Coccinelle) "2CV"
disponibilite (unique|
commandable_chez_constructeur) "unique"
etat (neuf|stock_invendu|occasion) "neuf">
<!ELEMENT Lot (Description,Prix)>
<!--ATTLIST Lot
ListePieces IDREFS #REQUIRED -->
```

On définit un élément « Lot » possédant des attributs « Description », « Prix » et « ListePieces » qui fait référence aux IDs définis dans les pièces.

```
<?xml version="1.0" ?>
<!DOCTYPE ListePieces SYSTEM "ListePieces_avec_lot.dtd">
<ListePieces>
<Piece reference="P834" modele="2CV" disponibilite="unique" etat="neuf">
<Description>Pot d'échappement</Description>
<Prix>150€</Prix>
</Piece>
<Piece reference="T563" modele="Mini" disponibilite="unique" etat="neuf">
<Description>sigle</Description>
<Prix>50€</Prix>
</Piece>
<Piece reference="M546" modele="2CV" disponibilite="unique" etat="occasion">
<Description>volant</Description>
<Prix>100€</Prix>
</Piece>
<Lot ListePieces="P834 M54"><Description> Lot pour 2CV </Description>
<Prix> 200 € </Prix></Lot>
</ListePieces>
```