

Feuille de style

xslt



Introduction

- Objectif de XML
 - structuration des informations
 - facilité d'écriture et d'interprétation
- Objectif des DTD
 - contraindre les balises xml
- Mais, bien que lisible le xml reste un langage informatique
- Les feuilles de styles
 - uniformiser la mise en page
 - utilisation du xls
- Format XSLT
 - syntaxe du xml
 - transformation d'un fichier XML

Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE constit SYSTEM "constit.dtd">
<constit>
  <code0AA_emetteur>Conseil constitutionnel</code0AA_emetteur>
  <code0DA_article_type>Article 61 alinéa 2</code0DA_article_type>
  <dates_seance>jeudi 17 janvier 2002</dates_seance>
  <code0EA_auteur_saisine>Députés Sénateurs</code0EA_auteur_saisine>
  <publication>
    <code0BA_date_jo>Journal officiel du 23 janvier 2002, p. 1503</code0BA_date_jo>
    <titre>
      <code1AA_titre>Loi relative à la Corse</code1AA_titre>
      <titrecommercial>Non conformité partielle</titrecommercial>
      <loi_traitee_nom_loi>"Loi relative à la Corse" nor_loi="INTX0000188L" num_loi="2002-92"/>
    </titre>
    <visas>
      <texte_decision>
        <code6AB_debut>Le Conseil constitutionnel a été saisi, dans les conditions prévues à l'a
        <code6BA_visa>Vu la Constitution ;$Vu l'ordonnance no 58-1067 du 7 novembre 1958 modifié
        <code6CA_motif>1. Considérant que les députés requérants mettent en cause l'article 1er
        <code6DA_dispositif>Décide : $$Art. 1er. - Sont déclarés contraires à la Constitution :$
      </texte_decision>
    </publication>
  </constit>
```

Règles de transformation (XSLT)

JORF du 23 janvier 2002 page 1503
texte n° 1

LOI

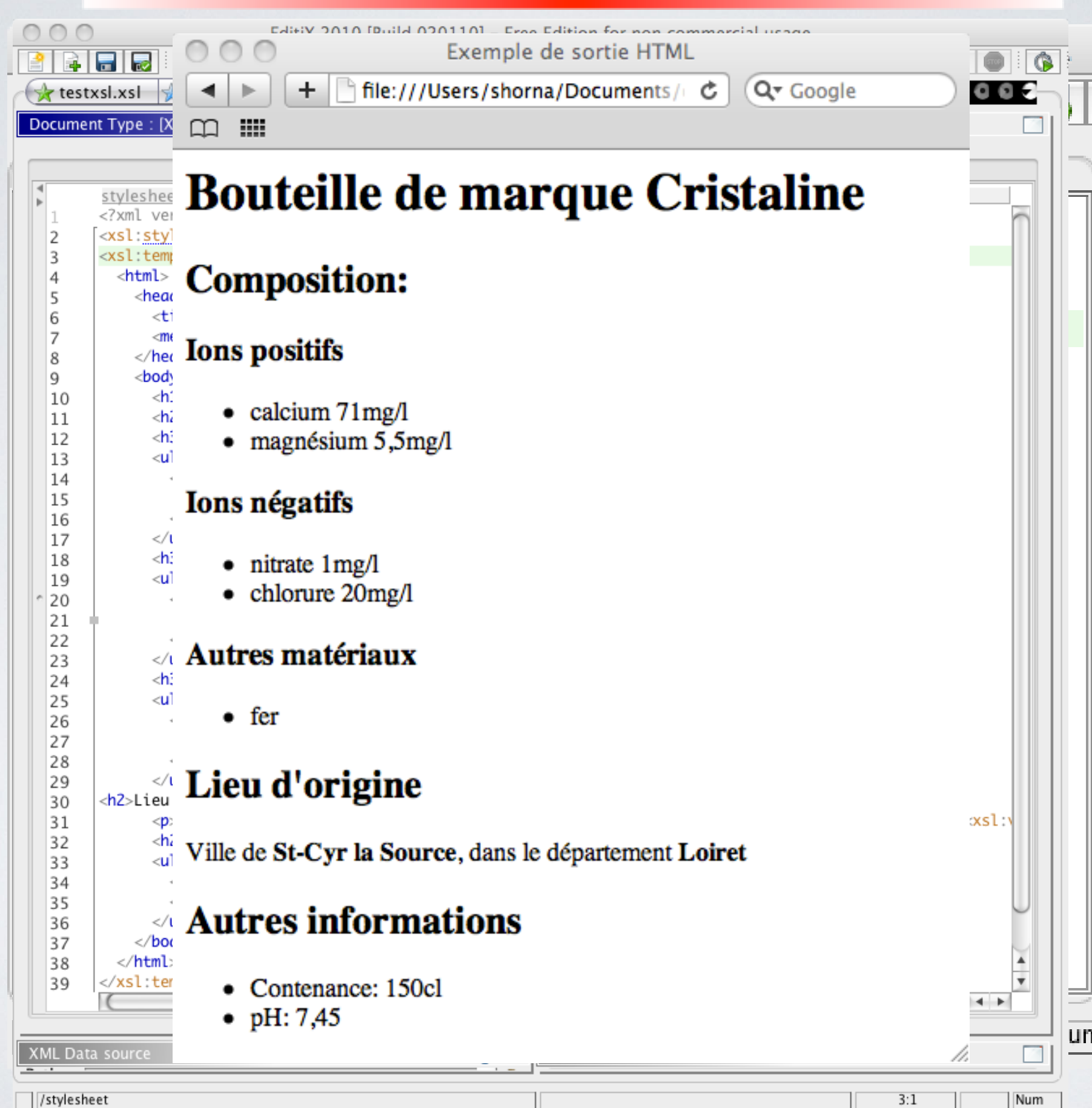
LOI n° 2002-92 du 22 janvier 2002 relative à la Corse (1)

NOR: INTX0000188L

ELI: <http://www.legifrance.gouv.fr/eli/loi/2002/1/22/INTX0000188L/jo/texte>

ELI: <http://www.legifrance.gouv.fr/eli/loi/2002/1/22/2002-92/jo/texte>

Introduction



Introduction

- XSL
 - eXtensive Stylesheet Language
 - ➡ langage extensible de feuille de style
- Recouvre 3 langages
 - XPath
 - ➡ moyen d'accéder à un noeud quelconque de l'arbre xml
 - XSLT
 - ➡ eXtensive Stylesheet LanguageTransformation
 - XSL-FO
 - ➡ eXtensive Stylesheet Language- Formating Objets
 - ➡ principalement utilisé pour la production finale en PDF
- Nous nous limiterons au 2 premiers aspects...

Structure

- XSLT
 - c'est un manipulateur d'éléments
 - transforme un document xml en un autre
- Fonctionnement
 - manipulation de modèles (templates)
 - les éléments xml sont remplacés (plus ou moins modifiés)
 - le modèle contient :
 - ➡ éléments...
 - ➡ attributs...
 - ➡ texte...
 - de remplacement

Structure

- Structure de base
 - un prologue
 - un élément : *xsl:stylesheet*
 - ➡ quelques attributs
 - ➡ déclaration d'espace de noms
 - ➡ numéro de version



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> (...) </xsl:stylesheet>

```

-U:**- test.xml All L3 (nXML Invalid)

- xsl:stylesheetStructure
 - élément racine
 - template de la racine `<xsl:template match= «/»>`

Instructions principales

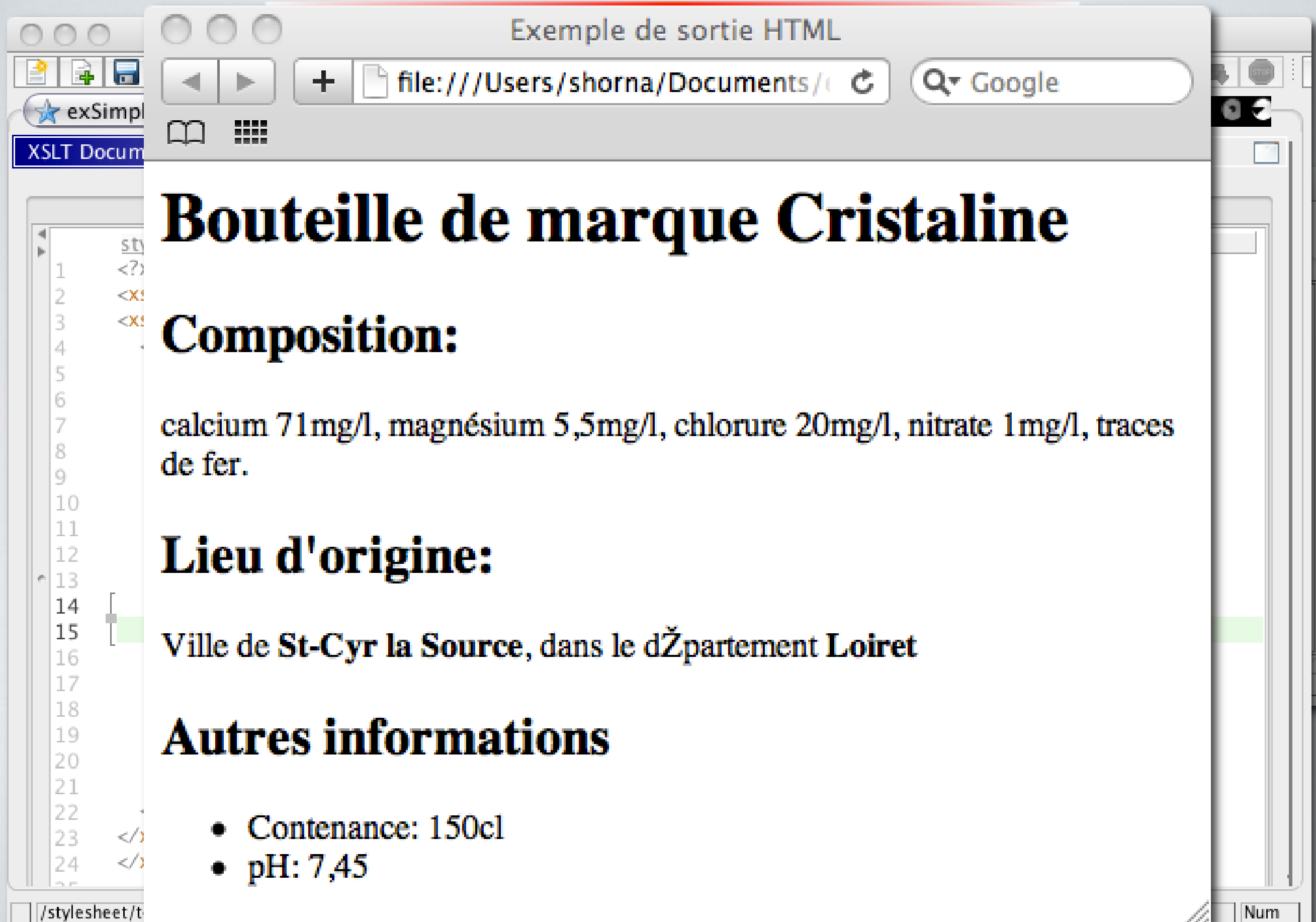
- Value-of : affiche la valeur d'un nœud ou d'un attribut.

```
<xsl:value-of select=« /livre/nom » /> <!-- valeur du sous-élément nom de l'élément livre>  
<xsl:value-of select=« //nom » /> <!-- valeur de tous les éléments nom où qu'ils soient>  
<xsl:value-of select=« @date » /> <!-- valeur de l'attribut date>  
<xsl:value-of select=« . » /> <!-- valeur de l'élément courant>  
<xsl:value-of select=« nom » /> <!-- valeur de l'élément nom>
```

- For-each : pour tous les nœuds

```
<xsl:template match="listePersonnes">  
  <ul>  
    <xsl:for-each select="Personne">  
      <li>  
        <xsl:value-of select="@nom"/>  
      </li>  
    </xsl:for-each>  
  </ul>  
</xsl:template>
```


Exemple simple



Exemple non linéaire

Editix 2010 [Build 020110] - Free Edition for non commercial usage

exSimple1.xml *bouteille1.xsl exSimple2.xml *bouteille2.xsl

XSLT Document

Main* Templates

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL"
3 <xsl:template match="/">
4   <html>
5     <head>
6       <title>Exemple de sortie HTML</title>
7       <meta http-equiv="Content-Type" content="text/html; charset=I
8     </head>
9     <body>
10      <h1>Bouteille de marque <xsl:value-of select="bouteille/marque" /></h1>
11      <h2>Composition:</h2>
12      <h3>Ions positifs</h3>
13      <ul>
14        <xsl:for-each select="bouteille/composition/ion_positif">
15          <li><xsl:value-of select="." /></li>
16        </xsl:for-each>
17      </ul>
18      <h3>Ions négatifs</h3>
19      <ul>
20        <xsl:for-each select="bouteille/composition/ion_négatif">
21          <li><xsl:value-of select="." /></li>
22        </xsl:for-each>
23      </ul>
24      <h3>Autres matériaux</h3>
25      <ul>
26        <xsl:for-each select="//autres_materiaux">
27          <li><xsl:value-of select="." /></li>
28        </xsl:for-each>
29      </ul>
30    <h2>Lieu d'origine</h2>
31    <p>Ville de <b><xsl:value-of select="bouteille/source/ville" /></b>, dans le département <b><xsl:value-of select="bouteille/source/departemen
32    <h2>Autres informations</h2>
33    <ul>
34      <li>Contenance: <xsl:value-of select="bouteille/contenance" /></li>
35      <li>pH: <xsl:value-of select="bouteille/ph" /></li>
36    </ul>
37  </body>
38 </html>
39 </xsl:template>
40 </xsl:stylesheet>
```

Rangement : ions positifs avant ions négatifs

Retour à la ligne