

# Les outils du XML



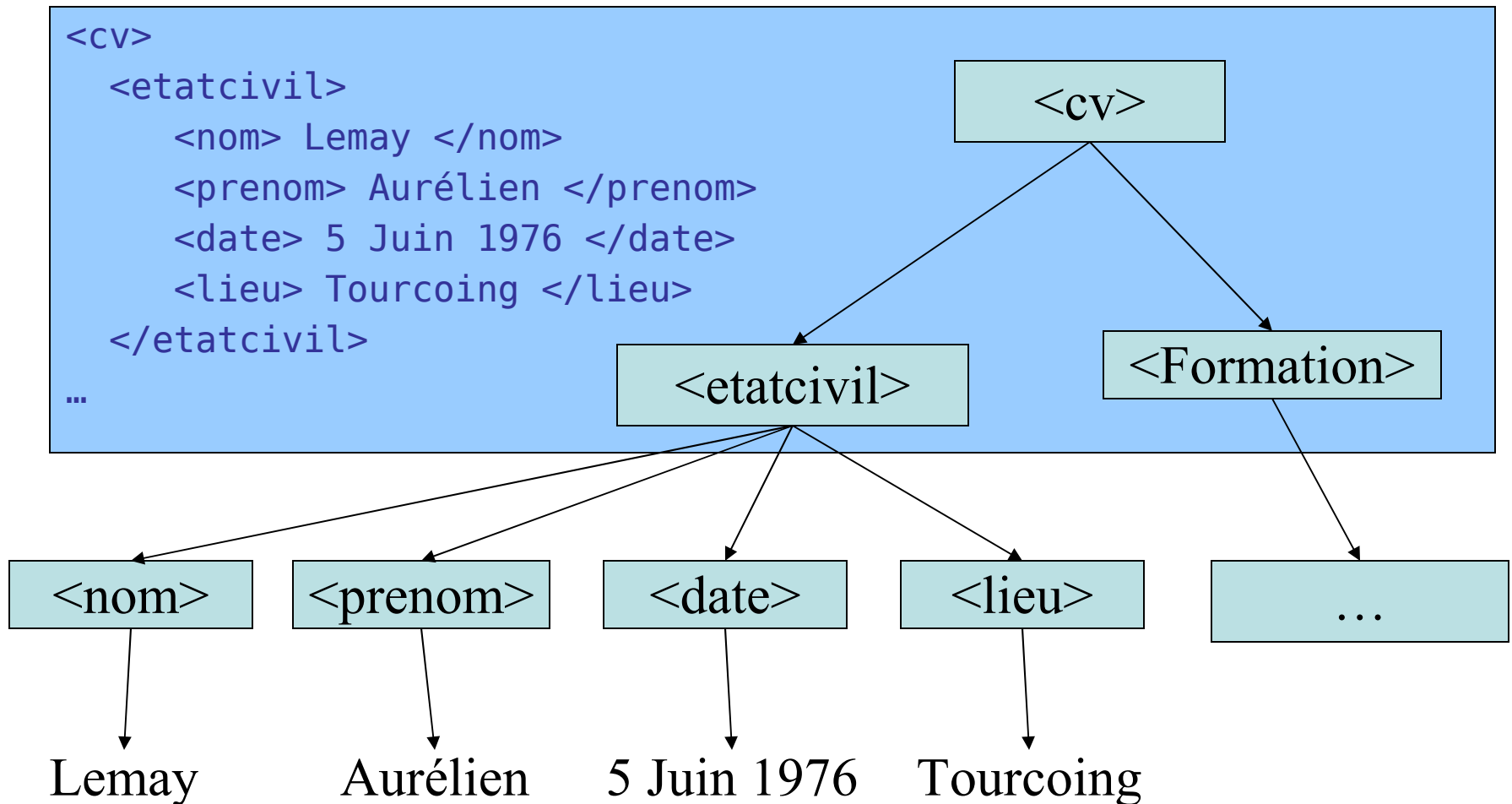
Xpath, XSL, XSLT, et autres trucs  
en X

# XPATH



Un outil de sélection de nœud

# Document XML : vue arborescente



# Sélection de nœud

- Le document XML contient des données
- On veut pouvoir les *extraire* automatiquement

Q : J'ai un CV, je veux trouver le nom de la personne, ou est-il ?

R : Il se trouve sous la balise <nom>, elle-même sous la balise <etatscivil>, sous <cv>

# Relations entre « nœuds »

```
<cv>  
  <etatcivil>  
    <nom>Lemay</nom>  
    <prenom>Aurélien</prenom>  
    <date>5 Juin 1976</date>  
    <lieu> Tourcoing </lieu>  
  </etatcivil>  
  ...
```

- <cv> est la **racine** du document
- <etatcivil> est le **père** de <nom>
- <nom> est le **fils** de <etatcivil>
- <prenom> est le **frère** de <nom>
- <nom> est un **descendant** de <cv>
- <cv> est un **ascendant** de <nom>

## XPATH : un langage de sélection de nœud

Q : J'ai un CV, je veux trouver le nom de la personne, ou est-il ?

R : obtenue par l'expression XPATH : /cv/  
etatcivil/nom

# Syntaxe XPATH

<i>balise</i>	Sélectionne tous les nœuds fils des nœuds ayant cette balise
/	Sélectionne à partir de la racine
//	Sélectionne à partir de n'importe où
.	Le nœud courant
..	Le père du nœud courant
@	Sélectionne un attribut

# Exemple XPATH

etatcivil	Les nœuds en dessous d'état civil
/cv	Le cv
etatcivil/nom	Le nom sous l'état civil
//nom	Tous les noms (où qu'ils soient)
cv//nom	Les noms qui sont descendants d'un nœud cv
//@photo	Sélectionne tous les attributs nommés photo



# Exercice (1)

## Préparation :

### Modifions le CV

- Toutes les dates : mettre uniquement une année
- Ajouter l'attribut photo dans etatcivil  
`<etatcivil photo="elvis.jpg">`

## Exercice (2)

Trouver les expressions XPATH  
correspondantes aux questions suivantes

- Quel est le prénom de la personne ?
- Quel sont les titres des formations ?
- Quel est le nom de la photo ?

# Correction

- Quel est le prénom de la personne ?  
**/cv/etatcivil/nom** ou **nom**
- Quel sont les titres des formations ?  
**//formation/titre**
- Quel est le nom de la photo ?  
**/cv/etatcivil/@photo**

# Les prédicats

Et si je veux faire des choses plus précises, dépendant d'une condition ?

Ex :

- quel est la formation suivie en 1999 ?
- Quelles sont les expériences professionnelles après 2002 ?
- Quelle est la première formation suivie ?

# Prédicats : exemples

<code>/cv/formation[1]</code>	La première formation
<code>/cv/formation[last()]</code>	La dernière formation
<code>/cv/formation[last()-1]</code>	L'avant dernière
<code>/cv/formation[position()]&lt;3]</code>	Les deux premières
<code>/cv/etatcivil[@photo]</code>	Les <b>etatcivil</b> qui ont une photo
<code>/cv/formation[date&gt;2000]</code>	Les formations après 2000
<code>/cv/formation[date&gt;2000]/titre</code>	Le titre des formations après 2000

## Exercice (3)

Trouvez les expressions XPATH correspondant aux requêtes suivantes :

- La dernière expérience professionnelle
- L'année de la dernière expérience professionnelle
- Le titre des formations ayant lieu à Lille

# Correction

- La dernière expérience professionnelle  
`/cv/experience[last()]`
- L'année de la dernière expérience professionnelle  
`/cv/experience[last()]/date`
- Le titre des formations ayant lieu à Lille  
`/cv/formation[lieu="Lille"]`

# Conclusion XPATH

- Xpath nous permet de rechercher des informations dans un document XML
  - Nous n'avons pas tout vu sur XPATH
- Voir [www.w3schools.com/xpath](http://www.w3schools.com/xpath) par exemple
- Il existe d'autres systèmes (Xquery, Xpointer, ...) mais Xpath est le plus courant



# **XSL, XSLT**



Transformation de documents XML

# XML

Différents formats XML (associés à une DTD) pour différents usages :

- Stockage de données
- Création de documents web
- Création de documents papiers
- Représentation des données (graphiques, dessins, ...)

# Pourquoi transformer ?

- Passer d'un format XML à un autre similaire  
Changement de logiciel par exemple
- Représenter une partie des données par un autre format

Ex : récupérer des données pour les afficher dans un graphique (format SVG)

- Utiliser un format plus adapté à l'utilisation qu'on fait des données

Ex : passer notre CV XML en XHTML pour le mettre sur Internet

# Transformation du CV

```
<CV>
  <etatcivil photo=moi.jpg>
    <nom>Lemay</nom>
    <prenom>Aurélien</prenom>
    <date>5 Juin 1976</date>
    <lieu>Tourcoing</lieu>
  </etatcivil>
  <formation>
    <titre>Doctorat</titre>
    <date>2002</date>
    <lieu>Lille</lieu>
    <mention>Très
    Honorable</mention>
  </formation>
  ...
```

```
<html>
  <head>
    <title>Aurélien Lemay
  </title>
  </head>
  <body>
    <div class="etatcivil">
      Aurélien Lemay ..
```

# Le CV

## Observons `cv.html`

- Il est bien mieux adaptés à l'affichage
- On en fait ce qu'on en veut

Q : Comment passer de `cv.xml` à `cv.html` ?

# Transformations XSL

- XSL : eXtensible Stylesheet Language

Un langage pour présenter (transformer) les fichiers XML

XSL : XML Stylesheet

XSL = XSLT (transformation)

- + Xpath (Sélection)

- + XSL-FO (formatage)

# XSLT : Abracadabra pour XML

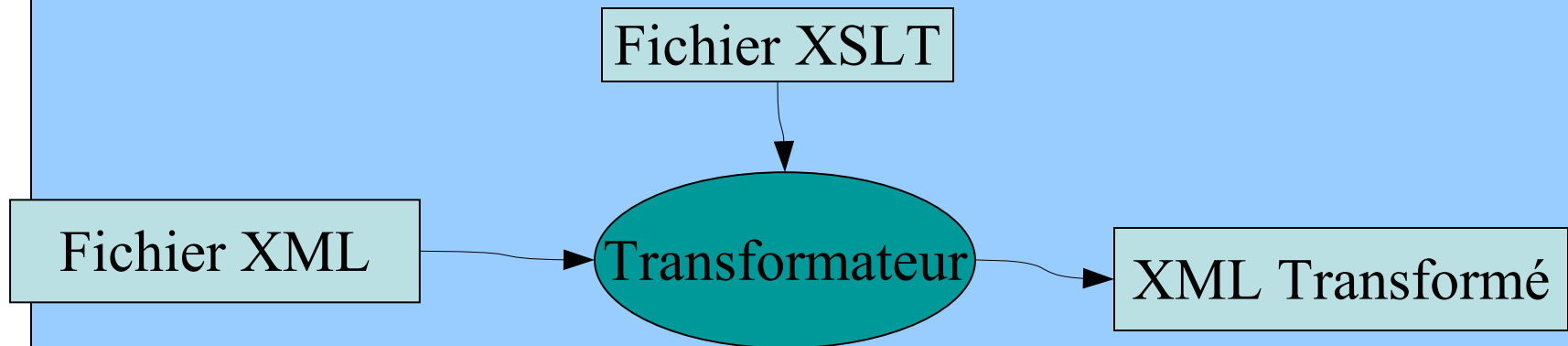
## XSLT : XSL Transformations

Langage permettant de *transformer* un fichier XML en un autre fichier XML

# Principe de Transformation

Un fichier XML : le point de départ

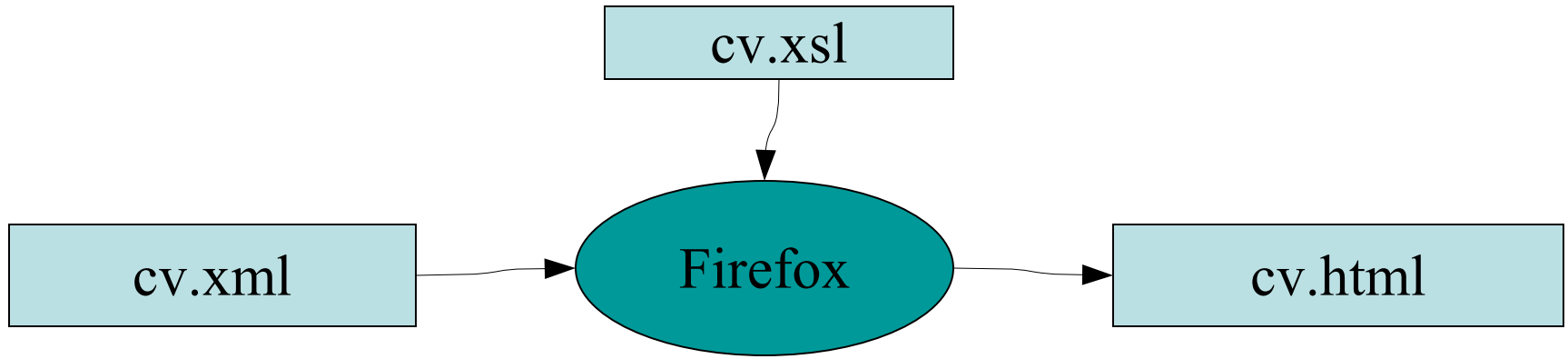
Un fichier XSLT : décrit la transformation



Un programme qui réalise la transformation



# Exemple



# (1) Lier le fichier XSLT

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<?xml-stylesheet type="text/xsl"  
  href="cdcatalog.xsl"?>  
<CV>  
  <etatcivil>  
  ...
```

Dans le fichier XML, on ajoute une ligne précisant le fichier XSLT à utiliser

## (2) Un premier fichier XSLT

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

- 1) Début classique d'un fichier XML
- 2) Une ligne pour préciser qu'il s'agit d'un fichier de transformation

```
<xsl:template match="...">
```

- Entre les guillemets : une expression Xpath
- Remplace le nœud correspondant par un certain contenu
- Fonctionne également pour tous le document si on met juste "/"

# 1er fichier XSLT

(debut)

```
<xsl:template match="/animal">
```

```
  <html> <head> ... </head>
```

```
  <body> René l'hippopotame    </body>
```

```
</html>
```

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

# xsl:value-of

```
<xsl:value-of select="(xpath)">
```

Est remplacé par le contenu du *premier* nœud référencé par l'expression

Si on est à l'intérieur d'un xsl-template, l'expression xpath *complète* celle du xsl-template

# Xsl:value-of : exemple

```
<xsl-template match="/animal/fiche">
  <div class="fiche">
    <DIV CLASS="NOM">
      <xsl:value-of select="fiche/nom" />
    </DIV>
    ...
  </xsl:template>
```

Récupère le contenu de /animal/fiche/nom

# Exercice

Complétons notre feuille de style pour récupérer toutes les infos de la partie fiche (sauf le sexe)



# Pour l'image ?

- On aurait envie d'écrire :  
`<img src=<xsl-value.... > >`
- Impossible ! Pas de balises imbriquées !
- Une syntaxe spéciale :  
`<IMG SRC="{fiche/photo}" />`

(entre les accolades : l'expression xpath)

# Exercice

Rajoutons l'image

# Xsl:for-each

- `Xsl:template` récupère un seul élément
- Lorsqu'il y a répétition : utiliser `xsl:for-each` permet de répéter le bloc pour tout élément qui répond à l'expression **xpath**

# Xsl:for-each : exemple

```
<xsl:template match="/cv">  
  <div class="formations">  
    <xsl:for-each select="formation">  
      <div class="formation">  
        <h1> <xsl:value-of select="titre"/></h1>  
      </div>  
    </xsl:for-each>  
  </div>  
</xsl:template>
```

# xsl:sort

Permet de faire un tri à l'intérieur d'un  
xsl:for-each

Ex :

```
<xsl:sort select="date"/>
```

# xsl:sort : exemple

```
<xsl:for-each select="formation">  
  <xsl:sort select="date">  
  <div class="formation">  
    <h1> <xsl:value-of select="titre"/></h1>  
  </div>  
</xsl:for-each>
```

# Exercice

- Complétons la feuille pour avoir les commentaires
- (pour l'instant, on met tous les commentaires en commentaires vétérinaire)

# xsl:if

- Met une sortie, seulement si une condition est remplie

```
<xsl:if test="date=1999">
```

...

```
</xsl:if>
```



# Exemple : Sexe des animaux

- Indiquez au bon endroit :

```
<xsl:if test="fiche/sexe = 'M' ">  
  <DIV CLASS="SEXE_M" />  
</xsl:if>  
<xsl:if test="fiche/sexe = 'F' ">  
  <DIV CLASS="SEXE_F" />  
</xsl:if>
```

# Exercice

- Idem pour les commentaires (Veterinaire ou Enclos)

Attention à respecter Majuscule / Minuscule dans votre test !

# Projet

- Une fois que la page web est terminée, créez la feuille XSLT permettant de transformer vos fichiers XML en fichier XHTML

# Mini-projet

