

Séance 4

Publier la TEI et Epidoc avec XSLT

TD Antiquités numériques - 2ème partie

Université Lumière Lyon 2, site Berges du Rhône

Mardi 9/12/2025 - 12h00-14h00

Visualiser un fichier XML dans un navigateur



Cascading Style Sheets (CSS)

- CSS est le langage utilisé pour coder la présentation et le style des pages web au format HTML (HyperText Markup Language).
- CSS peut aussi permettre de visualiser un fichier XML directement dans un navigateur.
- Il suffit d'ajouter l'instruction suivante au début du document (juste avant l'élément <TEI> par exemple (pour lui indiquer où trouver ce fichier)
`<xm1-styleSheet type="text/css" href="affichage-tei.css"?>`
-



Exercice 5 : consigne

1. Visualisez le corrigé de l'exercice 4 dans un navigateur à l'aide du fichier CSS 'affichage-tei.css'
2. Comparez avec la version en ligne
<https://inslib.kcl.ac.uk/irt2009/IRT308.html>
3. Regardez le fichier CSS et observer les choix de présentation.
Vous pouvez vous aider du document
'ressources/memo-css3.pdf'

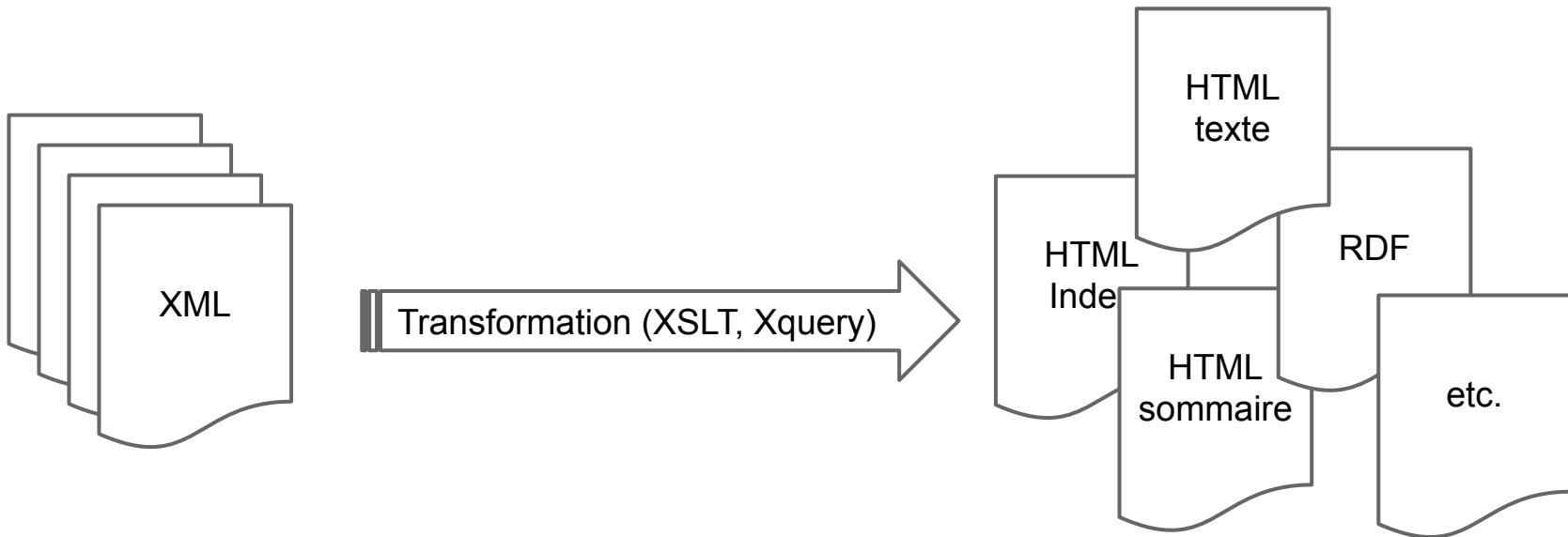
Comprendre la notion de “transformation” : introduction à XSLT



Pourquoi “transformer” l’arbre XML TEI ou EpiDoc ?

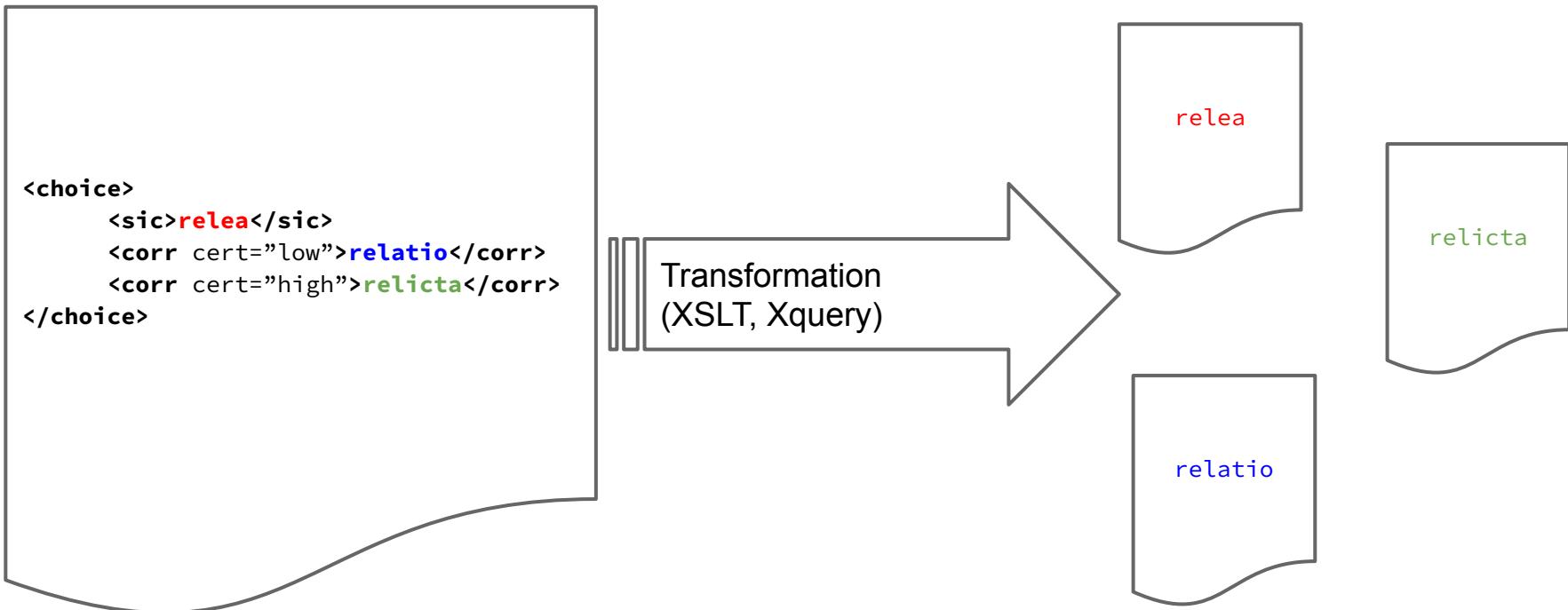
- XML ne “fait” rien, il se contente de décrire
- La publication de fichiers XML TEI ou EpiDoc requiert une étape de transformation permettant d’obtenir des formats de publication lisibles
 - HTML,
 - PDF
 - texte brut
 - ou bien d’autres arbres XML
- Les chaînes de publication basées sur XML établissent une séparation entre format d’encodage (données) et formats de publication (présentation des données) .

Single source publishing





“Single source publishing”

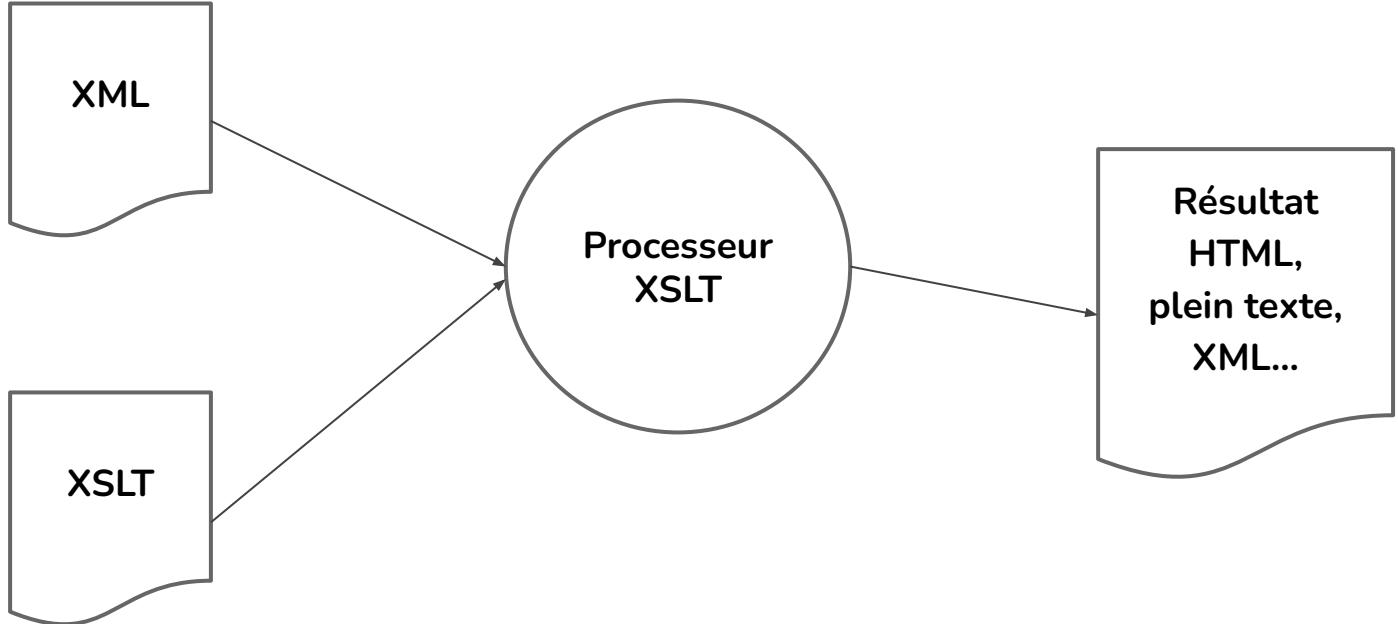




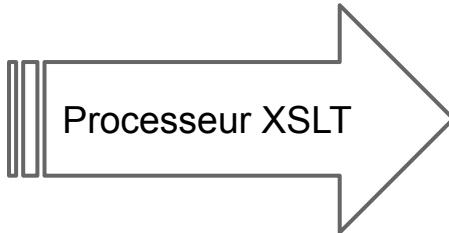
XSLT : un langage de transformation

- eXtensible Stylesheet Language Transformations
- Langage de transformation pour XML utilisé pour transformer la structure et le contenu d'un document XML en un autre format.
- XSLT est basé sur des règles de transformation (*templates*) qui définissent comment chaque élément XML doit être traité.

Principe de fonctionnement



```
<xml>  
<TEI>  
(...)  
<div>  
<head>Atelier  
pratique</head>  
<p>L'encodage XML/TEI,  
c'est facile !</p>  
</div>  
  
(...)  
</TEI>
```



```
<html>  
(...)  
<h1>Atelier pratique</h1>  
<p>L'encodage XML/TEI,  
c'est facile !</p>  
  
(...)  
  
</html>
```

XML

document en entrée
(input)

XSLT

Traitement

HTML

Résultat (output)



Nouvelles installations dans VsCode

Installer les plugin VSCode :

- [HTML Preview](#)
- [XSL Transform](#)

Installer également :

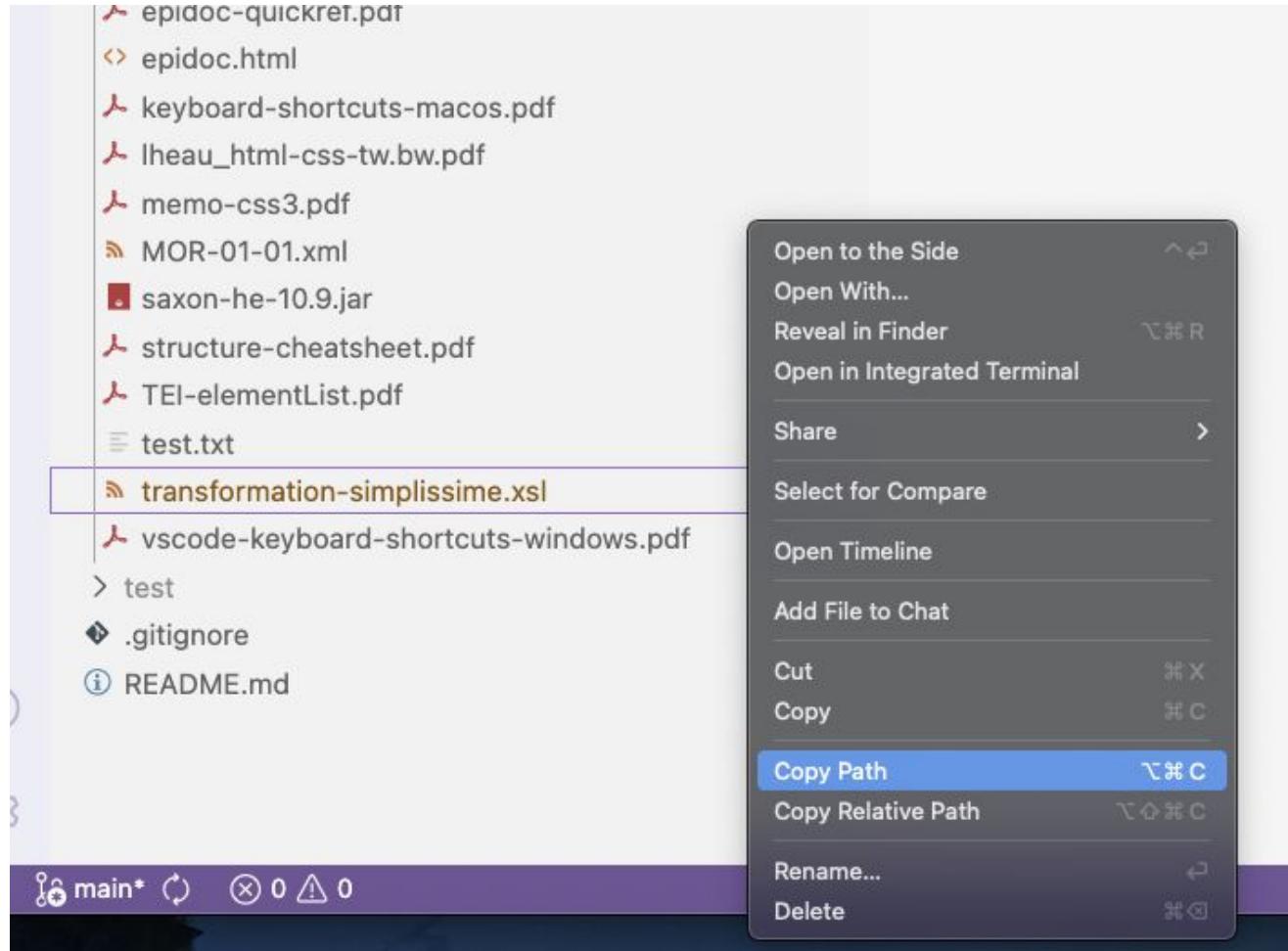
- [Java](#)
- Saxon (voir dossier ressources) : copiez le fichier `saxon-he-10.9.jar` dans un dossier de votre disque dur
 - PC : par exemple C:\saxon\saxon-he-10.9.jar
 - Mac : par exemple /Library/Java/Extensions/saxon-he-10.9.jar

Configurer le plugin XSL transform :

- settings > XSL:Processor en indiquant le chemin de Saxon ("saxon-he-10.9.jar")
- settings > XSL:stylesheet : indiquer le chemin de "ressources/transformation-simplissime.xsl"

Source : <https://francescagiannetti.com/vscode-for-tei/>

Voir aussi : <https://www.saxonica.com/html/documentation10/about/installationjava/installingjava.html>



Last synced: 0 secs ago

Xsl: Processor

Path to the Saxon XSLT processor (.jar)

/Library/Java/Extensions/saxon-he-10.9.jar

Xsl: Stylesheet

Path to active XSL / XSLT File

i-numerique-de-sources/ressources/transformation-simplissime.xsl



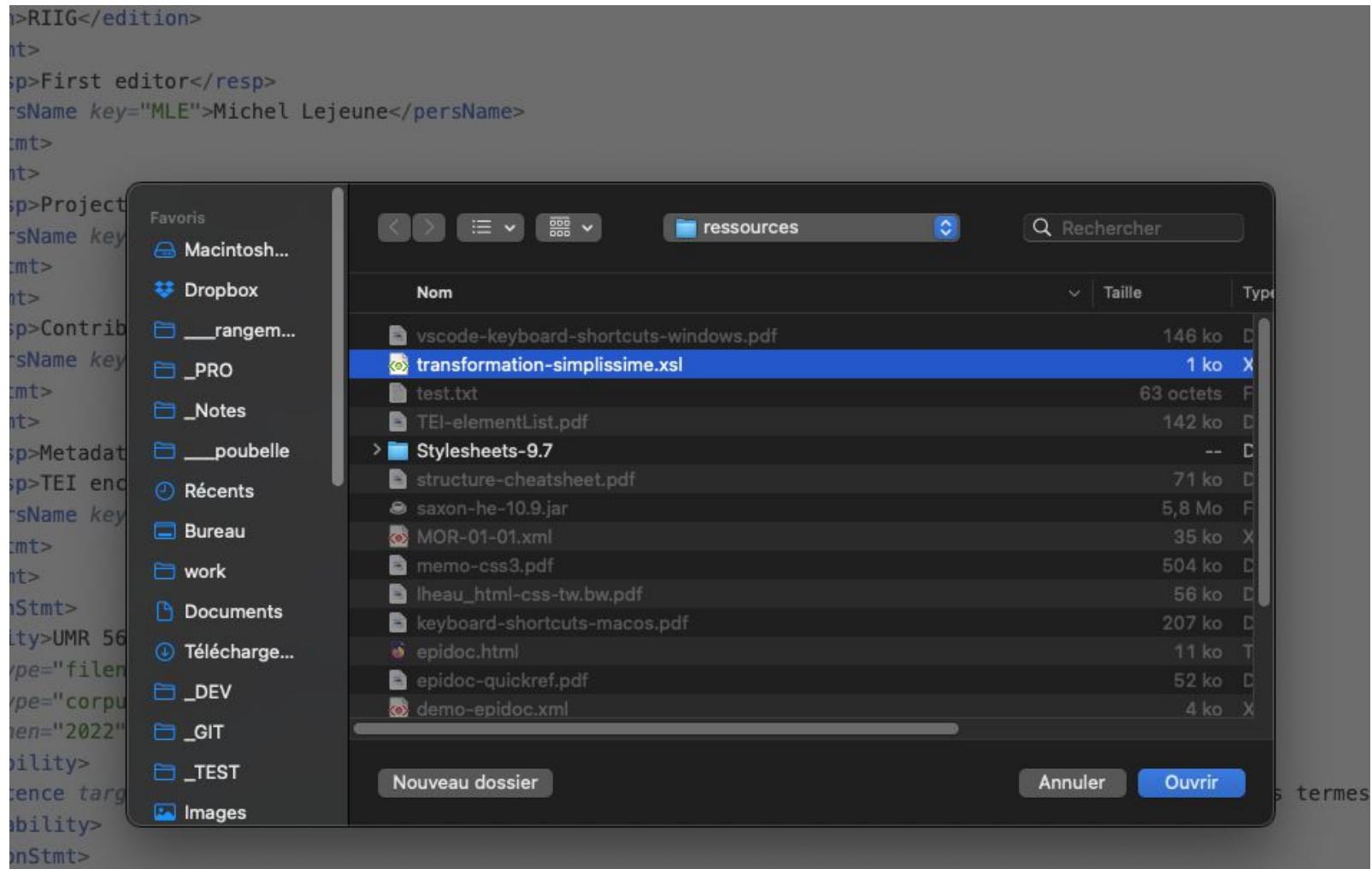
Lancer la transformation

- Ouvrez le fichier XML “ressources/MOR-01-01.xml”
- Ouvrez la palette de commandes avec MAJ + CMD + P (mac) / MAJ + CTRL + P (pc)
- Tapez “set” et sélectionnez “Set XSL transform” puis ouvrez le fichier “transformation-simplissime.xsl” pour indiquer quel fichier de règles XSLT vous souhaitez utiliser (même si on l'a indiqué avant dans la configuration des “settings”)
- Tapez “transform” jusqu'à ce qu'apparaisse ‘Transform document xsl.transform’ dans la liste puis sélectionner avec “entrée”
- Enregistrez le résultat au format texte (et non XML) : par ex. “test.txt”

MOR-01-01.xml ×

ressources > MOR-01-01.xml

```
1 <?xml-model href="http://epidoc.stoa.org/schema/latest/tei-epidoc.rng" type="application/xml" schematypens="http://www.tei-c.org/ns/1.0" xml:id="R296">
2 <TEI xmlns="http://www.tei-c.org/ns/1.0" xml:id="R296">
3   <teiHeader>
4     <fileDesc>
5       <titleStmt>
6         <title>Stèle votive de Plumergat</title>
7         <title type="corpus">RIIG</title>
8       </titleStmt>
9       <editionStmt>
10         <set>
11           Set XSL Stylesheet
12           Add Selection to Next Find Match
13           Add Selection to Previous Find Match
14           Chat: Add Selection to Chat workbench.action.chat.attachSelection
15           Chat: Add Selection to Chat github.copilot.chat.attachSelection
16           Chat: Chat Settings
17           Chat: Configure Tool Sets...
18           Chat: Reset Tool Confirmations
19           File: Reset Active Editor Read-only in Session
20           File: Set Active Editor Read-only in Session
21           File: Set Active Editor Writeable in Session
22           Help: Settings Sync
23           Merge Editor (Dev): Open Selection In Temporary Merge Editor
24           Merge Editor: Reset Choice for 'Close with Conflicts'
25         </set>
26       </editionStmt>
27     <publicationStmt>
28       <author>Move Selected Text Left
29       <idno type="uri">Move Selected Text Right
30       <idno type="uri">Preferences: Configure Language Specific Settings
31       <date when="2022">2022</date>
32     </publicationStmt>
33   </teiHeader>
```



... MOR-01-01.xml X

ressources > MOR-01-01.xml

```
1  <?xml-model href="http://epidoc.stoa.org/schema/latest/tei-epidoc.rng" type="application/xml"?
2  <TEI xmlns="http://www.tei-c.org/ns/1.0" xml:id="R296">
3      <teiHeader>
4          <fileDesc>
5              <titleStmt>
6                  <title>Stèle votive de Plumergat</title>
7                  <title type="corpus">RIIG</title>
```

>transform

Transform Document xsl.transform recently used ⚙

Transform Document xsl.transform

Transform to Camel Case other commands

Transform to Kebab Case

Transform to Lowercase

Transform to Pascal Case

Transform to Snake Case

Transform to Title Case

Transform to Uppercase

Transpose Characters around the Cursor similar commands

XML Tools: Convert text to XML (<> -> <>)

Refactor... R

Transpose Letters

Toggle Fold ⌘ K ⌘ L

Ask in Chat: transform ⌘ ⌘ I

29 <publicationStmt>
30 <authority>IMR 5607 – Ausonius/authority>

cours_intro-edition-numerique-de-sources

MOR-01-01.xml X

ressources > MOR-01-01.xml

```
1  <?xml-model href="http://epidoc.stoa.org/schema/latest/tei-epidoc.rng"
2  <TEI xmlns="http://www.tei-c.org/ns/1.0" xml:id="R296">
3      <teiHeader>
4          <fileDesc>
5              <titleStmt>
6                  <title>Stèle votive de Plumergat</title>
7                  <title type="corpus">RIIG</title>
8              </titleStmt>
9              <editionStmt>
10                 <edition>RIIG</edition>
11                 <respStmt>
12                     <resp>First editor</resp>
13                     <persName key="MLE">Michel Lejeune</persName>
14                 </respStmt>
15                 <respStmt>
16                     <resp>Project coordinator and contributor</resp>
17                     <persName key="CRD">Coline Ruiz Darasse</persName>
```

... ●

TRANSFORMATION XSLT Untitled-1

1 TRANSFORMATION XSLT

2

3 Nombre de références (balise w) : 18

S4 Fichiers

ORS 1 unsaved

New Text File
New File...
New Window
New Window with Profile
Open...
Open Folder...
Open Workspace from File...
Open Recent
Add Folder to Workspace...
Save Workspace As...
Duplicate Workspace
Save
Save As...
Save All
Share
Auto Save
Revert File
Close Editor
Close Folder [MK F]
Close Window

ressources > MCR-01-01.xml

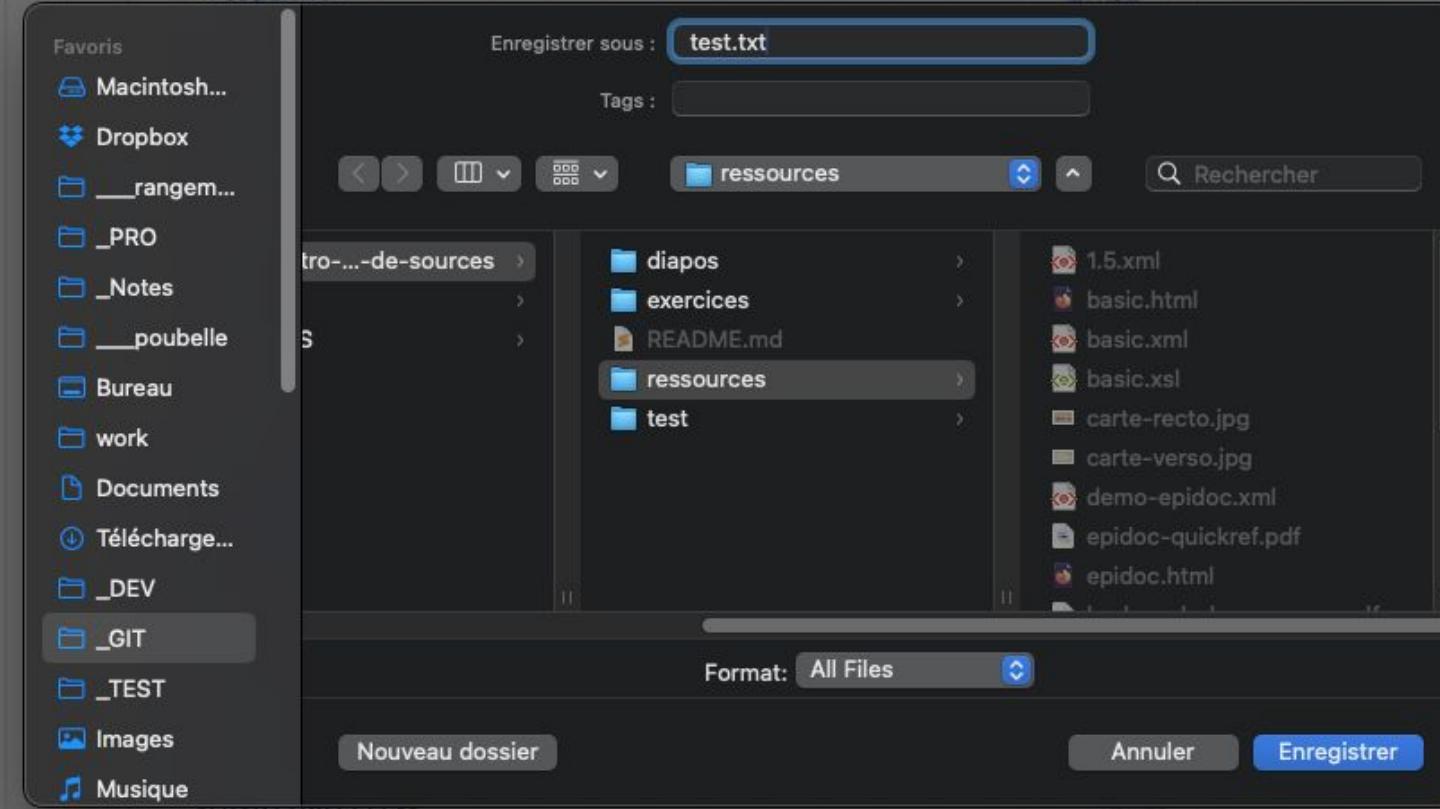
```
1  <?xml-model href="http://epidoc.stoa.org/schema/latest/tei-epidoc.rng"
2  ?>
3  <TEI xmlns="http://www.tei-c.org/ns/1.0" xml:id="R296">
4      <teiHeader>
5          <fileDesc>
6              <titleStmt>
7                  <title>Stèle votive de Plumerget</title>
8                  <title type="corpus">RIIG</title>
9              </titleStmt>
10             <editionStmt>
11                 <edition>RIIG</edition>
12                 <respStmt>
13                     <respFirst editor="resp">
14                         <persName key="MLE">Michel Lejeune</persName>
15                     </respFirst>
16                     <respStmt>
17                         <respProject coordinator and contributor="resp">
18                             <persName key="CRD">Coline Ruiz Darasse</persName>
19                         </respProject>
20                     </respStmt>
21                     <respStmt>
22                         <respContributor="resp">
23                             <persName key="HBT">Hugo Blanchet</persName>
24                         </respContributor>
25                     </respStmt>
26                     <respStmt>
27                         <respMetadata="resp">
28                             <respTEI encoding="resp">
29                                 <persName key="NCR">Nolwenn Chevalier</persName>
30                             </respTEI>
31                         </respMetadata>
32                     </respStmt>
33                     <availability>
34                         <silence target="http://creativecommons.org/licenses/by-nd/4.0/deed.fr"/>
35                     </availability>
36                 </editionStmt>
37                 <publicationStmt>
38                     <authority>UNR 5607 - Ausonius</authority>
39                     <idno type="filename">MOR-01-01</idno>
40                     <idno type="corpus-number">0</idno>
41                     <date when="2022">2022</date>
42                     <availability>
43                         <silence target="http://creativecommons.org/licenses/by-nd/4.0/deed.fr"/>
44                     </availability>
45                 </publicationStmt>
46                 <notesStmt>
47                     <note/>
48                 </notesStmt>
49                 <sourceDesc>
50                     <msDes>
51                         <msIdentifier>
52                             <country/>
53                             <region/>
54                             <settlement/>
55                             <repository/>
56                             <collection/>
57                             <idno type="invNo"/>
58                             <idno type="TM">172744</idno>
59                             <idno type="localID">L15</idno>
60                             <altIdentifier>
61                                 <idno type="localID">L-15</idno>
62                             </altIdentifier>
```

TRANSFORMATION XSLT Untitled-1

```
1  TRANSFORMATION XSLT
2
3  Nombre de références (balise w) : 18
```



```
<persName key="MLE">Michel Lejeune</persName>
</respStmt>
<respStmt>
```



```
</publicationStmt>
```

```
<notesStmt>
```

```
<note>
```



exercice 6 :

- Dupliquez “transformation-simplissime”
- Modifiez la règle XSLT pour compter cette fois le nombre de références bibliographiques (élément `<bibl/>`) se trouvant dans les blocs bibliographiques du fichier XML (les `div[@type='bibliography']`)...



Comment ça marche ?

- Le fichier XML en entrée est un arbre
- Le fichier XSLT contient une série de règles de transformation qui sont traitées “de bas en haut” en suivant la structure de l’arbre XML
- Un fichier XSLT est donc une série de “templates” (règles) XSLT non ordonnées (contrairement à des instructions de traitement que le programme devrait exécuter les unes après les autres). Ici l’ordre est donné par l’arbre XML. Le processeur XSLT descend dans l’arborescence et recherche les règles s’appliquant à chaque élément.
- Lorsqu’il n’y a pas de règle pour un élément parcouru, le processeur traite le noeud conformément aux règles par défaut (built-in). Typiquement : afficher le contenu textuel de l’élément.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/1999/xhtml"
  xpath-default-namespace="http://www.tei-c.org/ns/1.0"
  exclude-result-prefixes="#all">

  <xsl:output method="xml"/>

  <xsl:template match="/">
    <!--(...)-->
  </xsl:template>

  <xsl:template match="head">
    <!-- (...)-->
  </xsl:template>

  <xsl:template match="title">
    <!--(...)-->
  </xsl:template>

  <xsl:template match="note">
<!--  (...)-->
  </xsl:template>

</xsl:stylesheet>
```

```
<xsl:template match="/">
    <!-- Ecrit un titre dans le fichier suivi d'un saut de ligne -->
    <xsl:text>TRANSFORMATION XSLT

</xsl:text>
    <!-- Dénombrer les mots (balises w) en utilisant XPath      -->

    <xsl:text>Nombre de mots (balise w) : </xsl:text>
    <xsl:value-of select="count(//w)" />
</xsl:template>
```



XSL:templates

- un template = 1 règle de transformation.
- Chaque template traite un morceau de l'arbre XML.
- Le template est exécuté lorsque le processeur XSLT atteint le noeud concerné dans son parcours de l'arbre du fichier XML en entrée.
- Le template traite ce qui est indiqué dans l'attribut **@match** et qui est exprimé en **XPATH**.



Différencier les éléments dans la feuille de style XSL

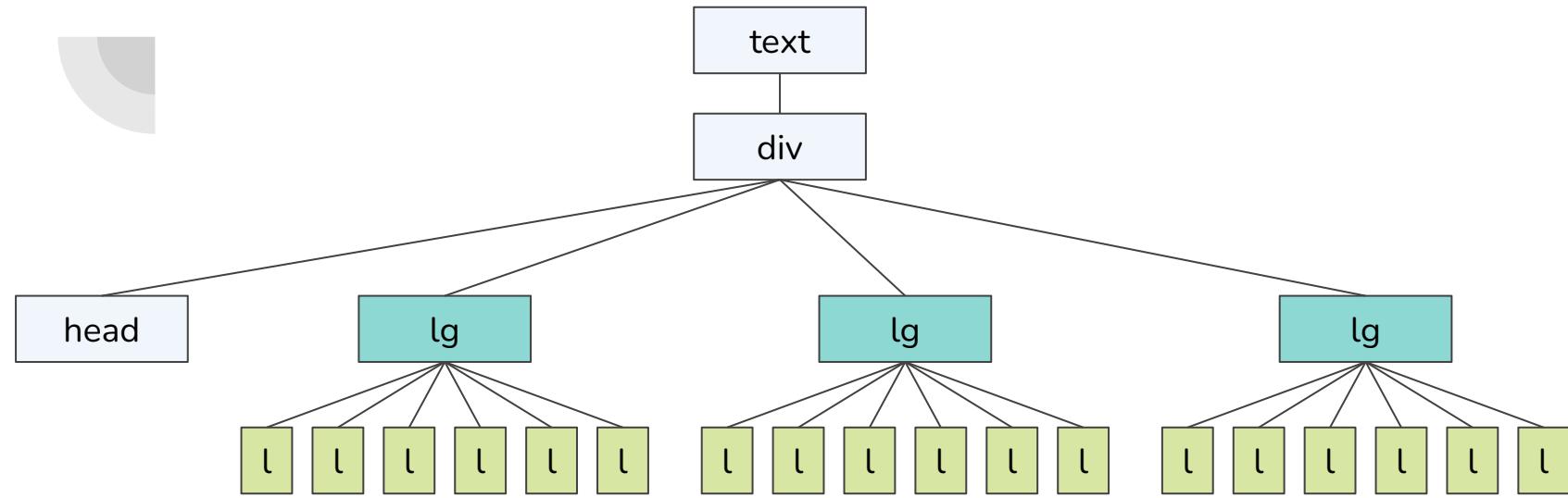
- On a affaire à **3 documents différents** :
 - le fichier XML (.xml)
 - le fichier XSLT contenant les templates / règles de transformation (.xsl)
 - le fichier résultat (par ex .txt ou .html)
- Dans le fichier XSL :
 - les noms d'éléments présents dans les attributs @match ou @select : correspondent aux éléments du fichier XML d'entrée
 - les éléments précédés de l'espace de nom “xsl:” : les éléments de transformation appartenant au “programme XSLT”
 - les éléments sans préfixe : correspondent aux éléments du format de sortie (si html)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.w3.org/1999/xhtml"
    xpath-default-namespace="http://www.tei-c.org/ns/1.0"
    exclude-result-prefixes="#all" >
```



Une transformation pas à pas

- Dossier ressources :
 - Transformer basic.xml avec basic.xsl et enregistrer dans basic.html



```
<text>
  <body>
    <div>
      <head>Chanson d'automne</head>
      <lg>
        <l> Les sanglots longs</l>
        <l>Des violons</l>
        <l>De l'automne</l>
        <l>Blessent mon coeur</l>
        <l>D'une langueur</l>
        <l>Monotone.</l>
      </lg>
      <lg>
        <l>Tout suffocant</l>
        <l>Et blême, quand</l>
        <l>Sonne l'heure,</l>
        <l>Je me souviens</l>
        <l>Des jours anciens</l>
        <l>Et je pleure</l>
      </lg>
      <lg>
        <l>Et je m'en vais</l>
        <l>Au vent mauvais</l>
        <l>Qui m'emporte</l>
        <l>Deçà, delà,</l>
        <l>Pareil à la</l>
        <l>Feuille morte.</l>
      </lg>
    </div>
  </body>
</text>
```

```
<xsl:template match="head">
  <h2><xsl:apply-templates/></h2>
</xsl:template>
```



Pour aller plus loin

- Cours vidéo XSLT de Syd Bauman et Martin Holmes sur la plateforme “Dariah Teach”
 - <https://teach.dariah.eu/course/view.php?id=32>

Utilisation des feuilles de style EpiDoc



Les feuilles de style EpiDoc

- Mises à disposition gratuitement sur <https://github.com/EpiDoc/Stylesheets>
- Maintenues par la communauté (une version par an environ)
- Paramétrables
 - via un fichier de paramètre ou des commandes directement passées au processeur XSLT
 - par exemple : intervalle de numérotation, type d'apparat critique, édition diplomatique ou interprétative...
- Paramètre '**\$leiden-style**' : différentes conventions d'affichage du texte (légères variations dans l'application des conventions de Leiden) :
 - ex 'panciera' (default), ila, london, petrae, seg, etc.
- Paramètre '**\$edn-structure**' : choix de mise en page et d'utilisation sélective du teiHeader



Installation

1. Téléchargez le jeu de fichiers XSLT depuis github
(<https://github.com/EpiDoc/Stylesheets>) et placez le fichier .zip de la version 9.7 dans le dossier “ressources”
2. Dé-comprimez le fichier .zip
 - Le point d’entrée pour une transformation vers html est “**start-edition.xsl**”
 - Le fichier de paramètres est “**global-parameters.xsl**”



Exercice : transformer un fichier EpiDoc à l'aide du jeu de feuilles de styles

- Ouvrez un fichier XML (par ex. le corrigé de l'exercice 4)
- Ouvrez la palette de VSCode, cherchez la commande “Set XSL Stylesheet”
- Indiquez “Stylesheets9.7/start-edition.xsl”
- Utilisez la commande “Transform document xsl.transform”
- Utilisez la commande “Format document” pour indenter le balisage html
- Enregistrez le résultat au format html “**exo-4.html**” par exemple
- Utilisez la commande “HTML: Open Preview” pour voir le résultat de la transformation dans un navigateur

EFES : un système de publication intégré pour EpiDoc

Welcome to EFES!

Now that you have EFES up and running, it's time to start building your project. Start with one of the options listed below and go from there.

Add your files

Put your EpiDoc XML files in webapps/ROOT/content/xml/epidoc and go to the [admin](#) and index them for searching. Then see how they are displayed ([Inscriptions](#)).

Customise the templates

Change the look of this site by modifying the templates in webapps/ROOT/assets/templates. The template for this page is home.xml, and it builds on the base template base.xml.

Read the documentation

The documentation and User Guide for EFES can be found on the project's [GitHub Wiki pages](#). Documentation for Kiln, the platform that EFES is based on, can be found both [online](#), and in source form, in the docs directory of your installation. It explains how you can modify everything about your project.



EFES

- *EpiDoc Front-End Services (EFES°)*
- Système de publication intégré pour EpiDoc
- Intègre les feuilles de styles XSLT que l'on vient de voir
- Intègre des index et un triplestore
- Accessible gratuitement sur github :
<https://github.com/EpiDoc/EFES>
- Installe un serveur web en local sur votre machine
 - Lancement en ligne de commande (cf. suivre la documentation)