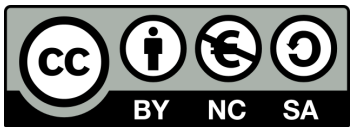
# Structuration des données et des documents

# Introduction à XML

J.B. Camps – jeudi 9 octobre 2014

M2 *Technologies numériques appliquées à l'histoire*

École nationale des chartes



# Objectifs de la séance

## I. Comprendre ce qu'est XML

A. Un premier document XML

B. Un premier langage XML

## II. Savoir quand/pourquoi/comment utiliser XML

# eXtensible (2) Markup Language (1)

1. Le ML de XML : *markup language*,
XML est un **langage à balise** ;
2. le X de XML : XML est ***eXtensible***,
*mais qu'est-ce que cela signifie* ?

École nationale des chartes

Jeudi 9 octobre 2014

# XML comme langage à balise

École nationale des chartes

Jeudi 9 octobre 2014

# XML comme langage à balise

On emploie *a priori* les italiques pour les locutions et termes empruntés à d'autres langues.

# XML comme langage à balise

On emploie *a priori* les italiques pour les locutions et termes empruntés à d'autres langues.

## Deux solutions possibles

On emploie `/ici commence un texte en italiques/` a priori `/ici finit un texte en italiques/` les italiques…

On emploie `/ici commence une locution étrangère/` a priori `/ici finit une locution étrangère/` les italiques…

# XML comme langage à balise

## On emploie *a priori* les italiques pour les locutions et termes empruntés à d'autres langues.

**TeX**

```
On emploie {\it a priori}
les italiques
```

**LaTeX**

```
On emploie \emph{a
priori} les italiques
```

**HTML**

```
On emploie <i>a
priori</i> les italiques
```

**RTF**

```
On emploie {\i a priori}
les italiques
```

**Wiki**

```
On emploie ''a priori''
les italiques
```

# XML comme langage à balise

On emploie *a priori* les italiques pour les locutions et termes empruntés à d'autres langues.

## Une solution en XML

```
On emploie <locutionÉtrangère>a priori</locutionÉtrangère> les italiques pour les locutions et termes empruntés à d'autres langues.
```

## En TEI

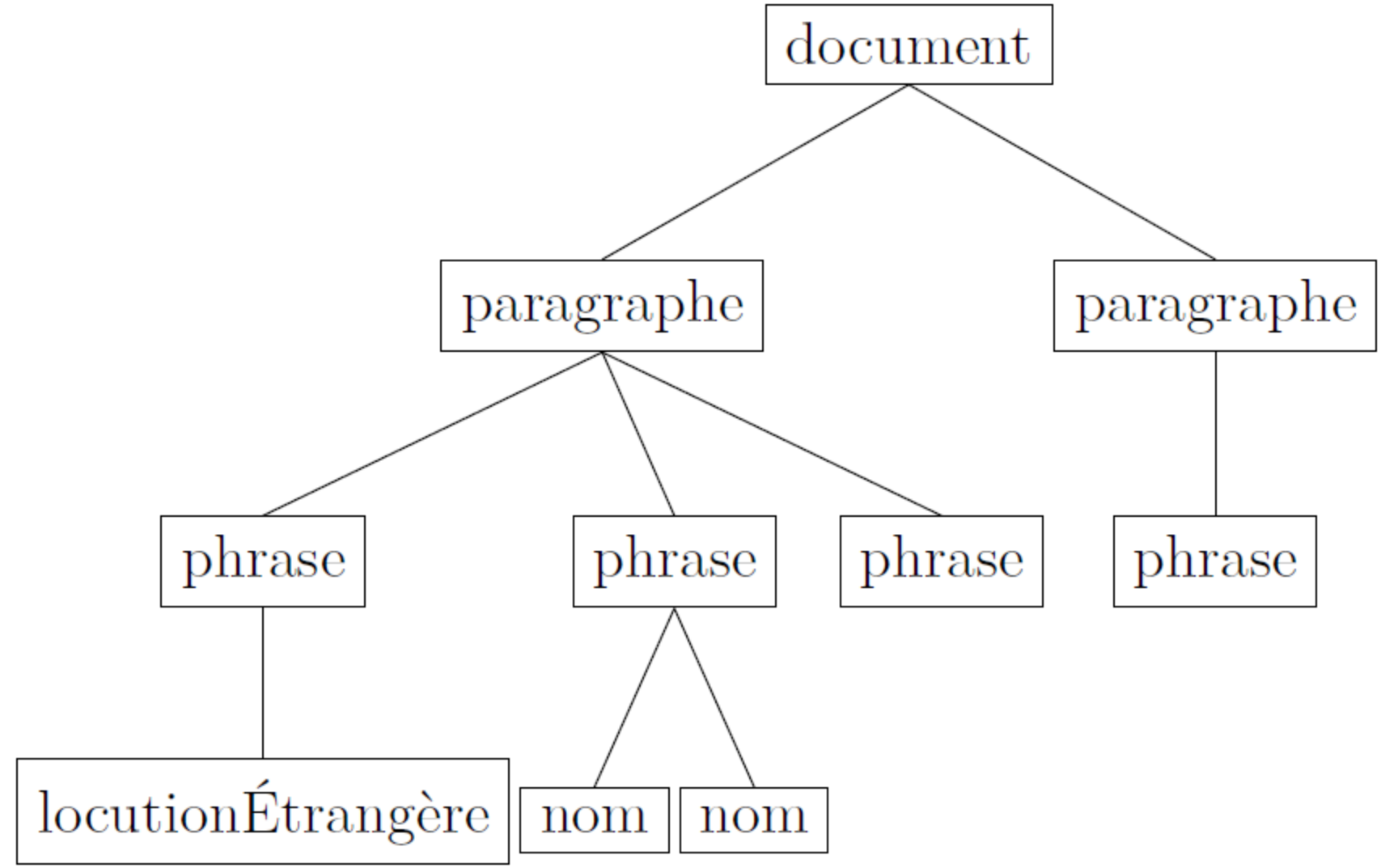
```
On emploie <foreign xml:lang="la">a priori</foreign > les italiques
```

# L'encodage sémantique et ses avantages

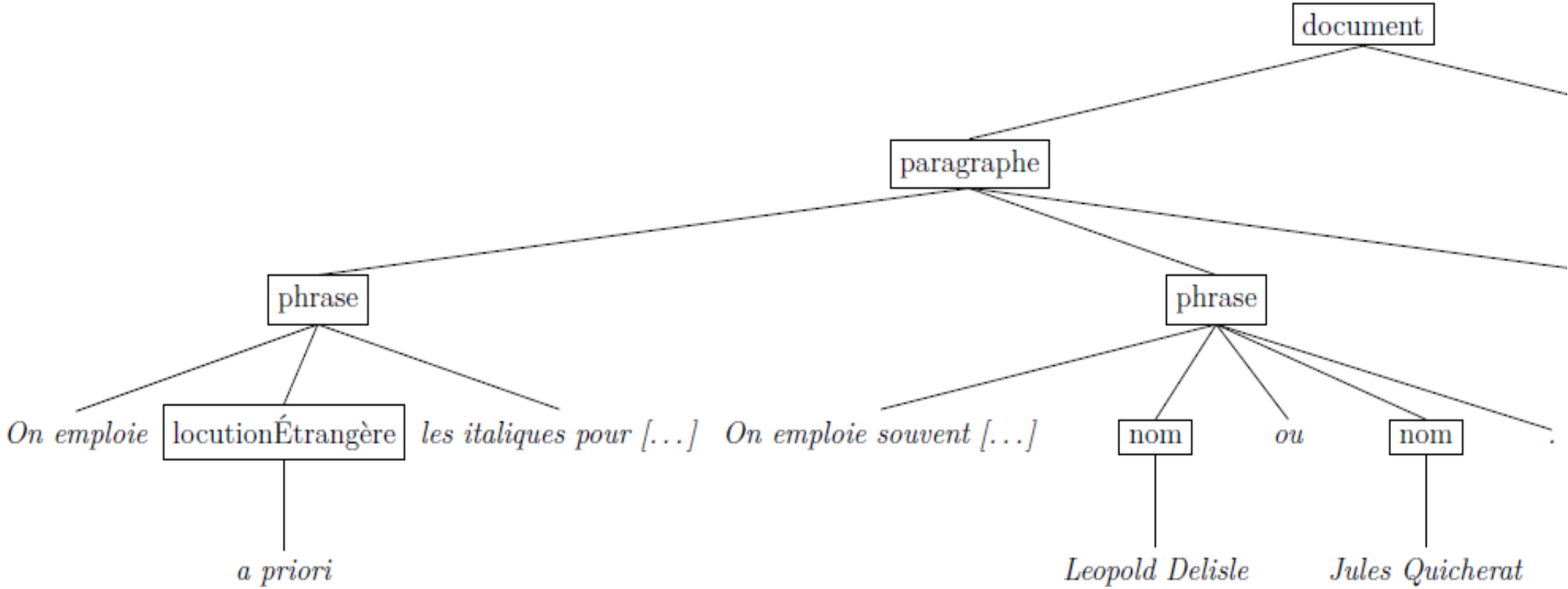
- Rend le **sémantisme naturel** d'un document **compréhensible par un ordinateur** et l'explicite
- Mise en forme **indépendante du document**, gérée à part et **(re)configurable**
- Permet une exploitation **utilisant le texte encodé comme base de données** (*data mining*, *information retrieval*, traitement automatique du langage, etc.)

# XML comme langage structuré

```
<document>
    <paragraphe>
        <phrase>On emploie <locutionÉtrangère>a
priori</locutionÉtrangère> les italiques pour les locutions
et termes empruntés à d’autres langues.</phrase>
        <phrase>On emploie souvent les petites capitales
pour les noms propres, comme <nom>Léopold Delisle</nom> ou
<nom>Jules Quicherat</nom>.</phrase>
        <phrase>On emploie en revanche généralement le gras
pour des raisons coupables.</phrase>
    </paragraphe>
    <paragraphe>
        <phrase>Un seconde paragraphe...</phrase>
    </paragraphe>
</document>
```


document
paragraphe
paragraphe
phrase
phrase
phrase
phrase
locutionÉtrangère
nom
nom

# Ou, plus exactement,



# Pas d'éléments entrecroisés

**Bien formé**

```
<auteur><nom>Dupont</nom></auteur>
```

**Pas bien formés**

```
<auteur><nom>Dupont</auteur></nom>
```

```
<poème type="douteux">
   <vers>À la fin de ce vers, je dis : <locutionÉtrangère>Sic</vers>
   <vers> transit gloria mundi</locutionÉtrangère></vers>
</poème>
```

# *eXtensible*

XML est un **métalangage**, c'est-à-dire qu'il permet de définir différents langages (des applications) avec chacun leur grammaire propre (par exemple TEI ou EAD ou XHTML) ;

contrairement à d'autres langages à balise, XML *ne propose pas un jeu de balise prédéfini*, mais **un ensemble d'éléments** et de **règles** sur ce que doit être un document **bien formé** ou un document **valide.**

# Les différents composants d'un document XML

Un document XML peut contenir :

- Des éléments (balises de début et de fin, contenu)
- Des attributs
- Des entités
- Sections CDATA
- Des instructions de traitement
- Des commentaires
- *et bien sûr du contenu textuel parsé (PCDATA)*

# Éléments

## Balise ouvrante et balise fermante, et contenu

contenu PCDATA

```
<nom>Dupont</nom>
```

contenu sous-éléments

```
<nom>
    <prénom>Jean</prénom>
    <patronyme>Dupont</patronyme>
</nom>
```

contenu mixte

```
<phrase>On emploie <locutionÉtrangère>a priori</locutionÉtrangère> les italiques... </phrase>
```

rien = élément vide

```
<lb/> (= <lb></lb>)
```

# Attributs

```
<nom type="personne">Dupont</nom>
<nom type='lieu'>Paris</nom>
<nom type='valeur"contenant"des"guillemets'>Autre nom</nom>

<personne naissance="1081-12-01" mort="1137-08-01">Louis VI</personne>

<lb raison="support"/>
```

# Entités

Comment écrire < ou & dans un document XML ?

ex. 6 < 7 , Laurel & Hardy , ...

# Entités

## Comment écrire < ou & dans un document XML ?

ex. 6 < 7 , Laurel & Hardy , …

6 < 7
Laurel & Hardy

Ou bien encore, en utilisant l'appel de caractère Unicode

6 < 7
Laurel & Hardy

# CDATA (≠PCDATA)

## Contenu entre **<![CDATA[** et **]]>**

```
<![CDATA[
#Ici, je peux mettre toutes les éperluettes que je veux et tous les chevrons, youpi
# &&&&&& <<<<<<<
# Je peux par exemple intégrer un script en Perl
while (1){
   print "Essayez de deviner le nombre compris entre 1 et 100 :\t ";
   my $tent = <STDIN>;
   if ($nombreADeviner == $tent){
         print "Bravo !\n";
         last
   }
   elsif ( ($tent == ' ') || ($tent =~ m/quitter|sortir/ )){
         print "Au revoir !\n";
         last
   }
   elsif ($tent < $nombreADeviner){
         print "Plus haut !\n"
   }
   elsif ($tent > $nombreADeviner){
         print "Plus bas !\n"
   }
}
]]>
```

# Instructions de traitement

Débutent par <? et se terminent par ?>
Permettent de faire appel à des applications extérieures

```
<?xml-stylesheet type="text/css" href="feuilledestyle.css"?>

<?xml-model href="ODD/out/Modele.rng" type="application/xml"
    schematypens="http://relaxng.org/ns/structure/1.0"?>
```

Également utilisées pour la **déclaration XML** (optionnelle en théorie mais **fortement recommandée** en pratique) au **début du document**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

# Commentaires

```
<!-- Commentaire
sur une ou plusieurs lignes
-->
```

# Un document XML **bien formé** (≠valide)

## Éléments

- Il y a un, et un seul, élément racine ;
- à chaque balise ouvrante correspond une balise fermante ;
- des éléments peuvent être imbriqués, mais pas entrecroisés ;

## Attributs

- valeurs des attributs entre guillemets (" ou ') ;
- pas deux attributs du même nom pour le même élément ;

## Autres

- pas de commandes et commentaires à l'intérieur d'une balise (ouvrante, fermante) même ;
- pas de caractère < ou & non échappé dans les données textuelles.

# Un document XML **bien formé** (≠valide)

## Un exemple minimal

**Ce document est bien formé**

```
<document/>
```

**Ce document est aussi bien formé**

```
<salutation>
  Bonjour
</salutation>
```

**Celui-ci aussi**

```
<BLUP>
  Bonjour
</BLUP>
```

# Un document XML **bien formé** (≠valide)

## Un exemple minimal

## Celui-là ne l'est pas

```
<BLUP>
  Bonjour
</blabla>
```

## Celui-là non plus

```
<BLUP>
  Bonjour
</BLUP>

<BLUP>
  Au revoir
</BLUP>
```

# Un exemple un peu moins minimal

Voir les sources TEI des documents publiés par les éditions électroniques de l'École, par exemple le formulaire d'Odart Morchesne

http://elec.enc.sorbonne.fr/morchesne/src/morchesne.xml

# N.B. : noms XML

*Attention, XML est sensible à la casse*

<balise/> ≠ <Balise/>

*Noms peuvent être composés de caractères alphanumériques, des soulignements (_), traits d'unions (-), et points (.)*

*Ils ne doivent pas débuter par un nombre, trait d'union ou point.*

*(caractères Unicode 2.0 pour XML 1.0)*

**Bien formés :** `<balise/>`, `<_balise/>`, `<autre_balise/>`, `<encore-une-autre_Balise.Farfelue.123/>`, `<ετικέτα/>`

**Pas bien formés :** `<1-balise/>`, `<.balise/>` , `<balise!/>`, `<autre/balise,mal,nommée/>`

# XML pour différents buts

*A minima*, on peut distinguer :

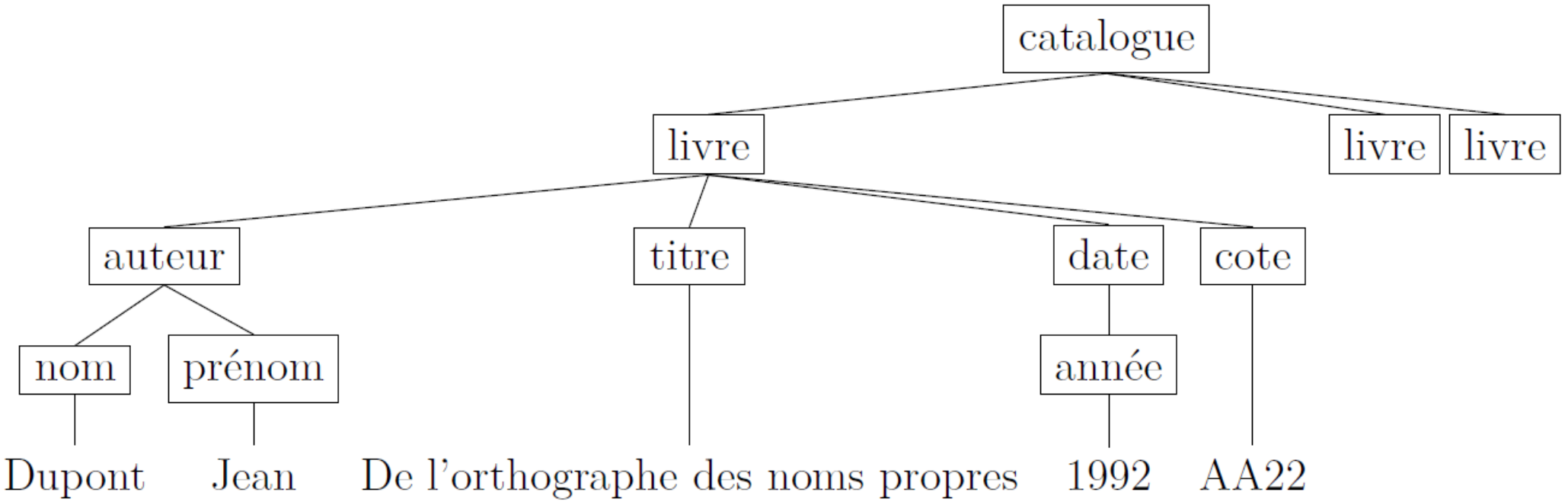
- XML pour des documents, une narration ;
- XML pour des bases de données

# Pour un document/narration

```
<document>
    <paragraphe>
        <phrase>On emploie <locutionÉtrangère>a
priori</locutionÉtrangère> les italiques pour les locutions
et termes empruntés à d’autres langues.</phrase>
        <phrase>On emploie souvent les petites capitales
pour les noms propres, comme <nom>Léopold Delisle</nom> ou
<nom>Jules Quicherat</nom>.</phrase>
        <phrase>On emploie en revanche généralement le gras
pour des raisons coupables.</phrase>
    </paragraphe>
</document>
```

# Pour des bases de données

```
<catalogue>
    <livre>
        <auteur><nom>Dupont</nom><prénom>Jean</prénom></auteur>
        <titre>De l'orthographe des noms propres</titre>
        <date><année>1992</année></date>
        <cote>AA22</cote>
    </livre>
    <livre>
        <auteur><nom>Chartiste</nom><prénom>Jules</prénom></auteur>
        <titre>Repertorium sermonum sancti Bernardi Claravalensis</titre>
        <date><année>1892</année></date>
        <cote>AA23</cote>
    </livre>
    <livre>
        <auteur><nom>Martin</nom><prénom>Jacques</prénom></auteur>
        <titre>tlhIngan Hol vIghoj ! Le klingon en seulement 122 leçons</titre>
        <date><année>2012</année></date>
        <cote>AA24</cote>
    </livre>
</catalogue>
```


catalogue
livre
livre
livre
auteur
titre
date
cote
nom
prénom
année
Dupont
Jean
De l'orthographe des noms propres
1992
AA22

# T.P. : Mon premier document XML

À faire : réinventez le fil à couper le beurre en proposant un encodage XML qui rende compte du sémantisme du document *document.pdf*

École nationale des chartes

Jeudi 9 octobre 2014

# XML et ses grammaires

## Qu'est-ce qu'un document XML **valide** ?

# XML et ses langages

Qu'est-ce qu'un document XML **valide** ?

Un document XML **valide** est un document qui se conforme à une **grammaire** préétablie, contenue dans une DTD ou un schéma.

# La syntaxe d'une DTD

## La déclaration de DTD

DTD externe

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE nomÉlémentRacine SYSTEM "madtd.dtd">

<!DOCTYPE ead PUBLIC" +//ISBN 1-931666-00-8//DTD ead.dtd
(Encoded Archival Description (EAD) Version 2002)//EN"
"../shared/ead/ead.dtd"
```

DTD interne

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE nomÉlémentRacine [
   <!ELEMENT nomÉlémentRacine (#PCDATA)>
   Définition des autres éléments...
]
```

DTD externe avec sous-ensembles internes

```
<!DOCTYPE document SYSTEM "madtd.dtd" [
   <!ELEMENT elementNonDecritDansmadtd.dtd (#PCDATA)>
]
```

# Vérifier la validité d'un document

- **Dans <oXygen/>**

- **via xmllint**

```
$ xmllint --valid mondocument.xml
```

# Déclaration d'élément

<!ELEMENT *nomdelelement* (*definitionducontenu*) >

**Contenu peut être :**
- données textuelles (#PCDATA)
- sous éléments (à définir également), qui peuvent être
    - obligatoires
    - optionnels ( * zéro ou plusieurs ; ? zéro ou un ; + un ou plusieurs)
- vide (EMPTY)

**Opérateurs**
- ou : |
- et (dans cet ordre) : ,
- groupe d'éléments : ()

# Pour des données structurées…

```
<catalogue>
    <livre>
        <auteur><nom>Dupont</nom><prénom>Jean</prénom></auteur>
        <titre>De l'orthographe des noms propres</titre>
        <date><année>1992</année></date>
        <cote>AA22</cote>
    </livre>
    <livre>
        <éditeur><nom>Chartiste</nom><prénom>Jules</prénom></éditeur>
        <titre>Repertorium sermonum sancti Bernardi Claravalensis</titre>
        <date><année>1892</année></date>
        <réédition><année>1963</année></réédition>
        <cote>AA23</cote>
    </livre>
    <livre>
        <auteur><nom>Martin</nom><prénom>Jacques</prénom></auteur>
        <titre>tlhIngan Hol vIghoj ! Le klingon en seulement 122 leçons</titre>
        <date><année>2012</année></date>
        <cote>AA24</cote>
        <commentaire>Exemplaire manquant</commentaire>
    </livre>
</catalogue>
```

# Déclaration d'élément

<!ELEMENT *nomdelelement* (*definitionducontenu*) >

**Contenu peut être :**
- données textuelles (#PCDATA)
- sous éléments (à définir également), qui peuvent être
    - obligatoires
    - optionnels ( * zéro ou plusieurs ; ? zéro ou un ; + un ou plusieurs)
- vide (EMPTY)

**Opérateurs**
- ou : |
- et (dans cet ordre) : ,
- groupe d'éléments : ()

<!ELEMENT catalogue (livre+) >
<!ELEMENT livre (auteur,titre,date,cote) >
<!ELEMENT titre (#PCDATA) >

# Exercice : écrivez la DTD de ce document

```
<catalogue>
    <livre>
        <auteur><nom>Dupont</nom><prénom>Jean</prénom></auteur>
        <titre>De l'orthographe des noms propres</titre>
        <date><année>1992</année></date>
        <cote>AA22</cote>
    </livre>
    <livre>
        <éditeur><nom>Chartiste</nom><prénom>Jules</prénom></éditeur>
        <titre>Repertorium sermonum sancti Bernardi Claravalensis</titre>
        <date><année>1892</année></date>
        <réédition><année>1963</année></réédition>
        <cote>AA23</cote>
    </livre>
    <livre>
        <auteur><nom>Martin</nom><prénom>Jacques</prénom></auteur>
        <titre>tlhIngan Hol vIghoj ! Le klingon en seulement 122 leçons</titre>
        <date><année>2012</année></date>
        <cote>AA24</cote>
        <commentaire>Exemplaire manquant</commentaire>
    </livre>
</catalogue>
```

# La DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT catalogue (livre+) >
<!ELEMENT livre ( (auteur | éditeur)+,titre,date,réédition*,cote,commentaire?) >
<!ELEMENT auteur (nom,prénom) >
<!ELEMENT éditeur (nom,prénom) >
<!ELEMENT nom (#PCDATA) >
<!ELEMENT prénom (#PCDATA) >
<!ELEMENT titre (#PCDATA) >
<!ELEMENT date (année) >
<!ELEMENT réédition (année) >
<!ELEMENT année (#PCDATA) >
<!ELEMENT cote (#PCDATA) >
<!ELEMENT commentaire (#PCDATA) >
```

# Autres types d'éléments

**Pour des données « narratives »**
Lorsqu'un élément possède **un contenu mixte** (texte parsé et autres éléments), **la seule syntaxe possible est la suivante :**

**<!ELEMENT monélément (#PCDATA | elementenfantoptionnel | autrelementenfantoptionnel | etc.)* >**

**Pour un élément vide**
**<!ELEMENT lb EMPTY>**

**Pour tout et n'importe quoi (à ne pas faire)**
**<!ELEMENT monÉlémentImprécis ANY>**

# Attributs

## Déclarés par la déclaration <!ATTLIST >

<!ATTLIST élément attribut type #ParDéfaut>

### Les types de données d'attribut

- CDATA ;
- nom XML (NMTOKEN ou NMTOKENS)
- nom XML unique (ID) ou déclaré ailleurs (IDREF, IDREFS)
- entité (ENTITY ou ENTITIES)
- énumération des valeurs possibles (valeur 1| valeur2|etc.)

# Attributs

Déclarés par la déclaration <!ATTLIST >

<!ATTLIST élément attribut type #ParDéfaut>

**Les types d'attribut et valeurs par défaut**

- #IMPLIED : optionnel, pas de valeur par défaut
- #REQUIRED : obligatoire, pas de valeur par défaut
- #FIXED "valeur" : valeur fixe non modifiable
- "valeur par défaut"

# Quelques exemples

```
<!ATTLIST lb raison CDATA #IMPLIED >

<!ATTLIST nom type (personne|lieu) #REQUIRED>

<!ATTLIST catalogue   version CDATA #FIXED "1.0"
                      auteur  CDATA "JB Camps"
                    identifiant ID    #REQUIRED
>
```

# Entités

```
<!ENTITY entité "contenu">
<!ENTITY ENC "École nationale des chartes">
```

&ENC; sera développé par le parseur en « École nationale des chartes »

## Entités externes

parsées

```
 <!ENTITY documentxmlàinsérer SYSTEM
"/home/user/document/doc.xml">
```

non parsées

```
<!ENTITY monimage SYSTEM "image.jpg" NDATA jpeg >
<!NOTATION jpeg SYSTEM "image/jpeg">
```

# « Raccourcis » (Entités paramètres)

<!ENTITY % nom "contenu" >

Puis est appelée par %nom;

Un exemple

<!ELEMENT auteur (nom,prénom) >
<!ELEMENT éditeur (nom,prénom)

<!ENTITY % noms "nom,prénom" >

<!ELEMENT auteur (%noms;) >
<!ELEMENT éditeur (%noms;) >

# Les DTD et leurs limitations

## Limitations

- pas de **typage précis** du contenu des éléments (chaîne de caractères de texte, nombre entier, etc.) ou de leur sens (date, heure, nom, etc.)
- peu de **contraintes sur les éléments** (nombre d'occurrences, élément racine,...)
- Et enfin, les DTD en elles-mêmes ne sont **pas écrites en XML**

## Schémas

- W3C XML Schema, RelaxNG, ...
- ODD (TEI)

# T.P. : Ma première grammaire XML

**Objectif :** reprendre le document XML bien formé que vous avez créé tout à l'heure, et en écrire une DTD qui permette de le valider.

École nationale des chartes

Jeudi 9 octobre 2014

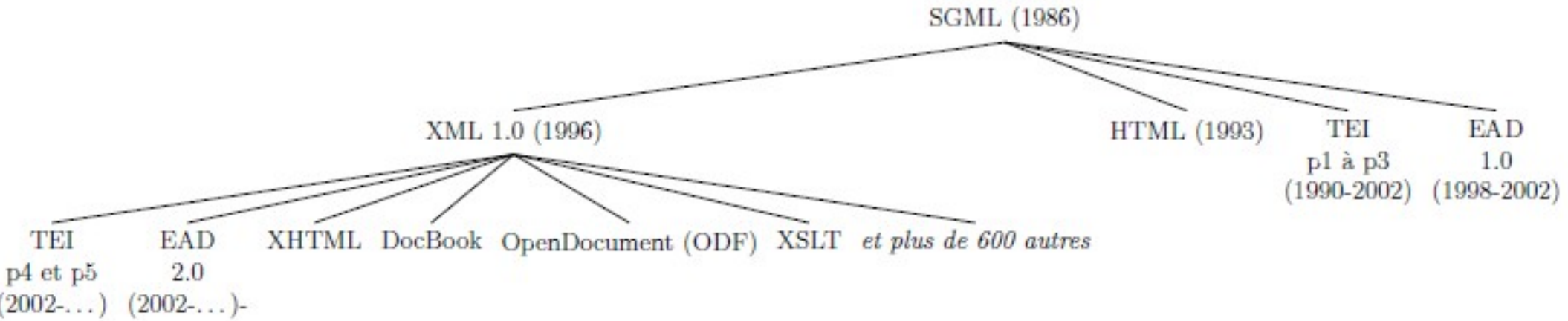
# I. Comprendre ce qu'est XML

# II. Savoir quand/pourquoi/comment utiliser XML

École nationale des chartes

Jeudi 9 octobre 2014

# Historique : la naissance de XML




Pour une comparaison SGML / XML, voir :
*World Wide Web Consortium, Comparison of SGML and XML*, 15-December-1997, en ligne : http://www.w3.org/TR/NOTE-sgml-xml-971215

Pour la spécification SGML :

# Ex. de code SGML (TEI p3)

```
<div type=chapter>
 <P id="p1">This could be a TEI encoded<LB>
 single-paragraph chapter<LB>
 whose line breaks have been retained.
 </P>
</div>
```

cf. *teibyexample*

# Ce que peut faire XML (ou pas)

- un langage pour des documents textuels, créé pour des documents « narratifs »
- un langage également très utilisé, sans que ce soit sa destination initiale, pour des données structurées sous forme de bases

Mais XML n'est pas
- un langage de programmation
- un système de gestion de bases de données relationnelles

# Le Choix de XML pour un projet

- Pérennité et interopérabilité
- multiplicité des usages
  - mise en forme
  - exploitation
  - réutilisation
- contenu lisible par l'humain autant que par l'ordinateur
- existence de nombreux dialectes et applications
- environnement de travail

# XML dans un environnement de travail

des outils permettant

- de créer des feuilles de style (CSS, XSL-FO)
- de localiser ou identifier des parties de document (XPath) ou les lier (XLink)
- de combiner plusieurs documents (XInclude)
- d'utiliser des documents XML comme des bases de données et de les interroger (via XQuery)
- ....

ainsi que de puissants outils de transformation (**XSLT**)

# Quelques langages bien connus

- TEI, pour l'édition scientifique
- DocBook, pour la documentation
- XHTML, pour les pages Web
- OpenDocument Format, pour des documents de traitement de texte

*Mais aussi pour*

- la description archivistique (EAD)
- bibliographique (MarcXML, MODS,...)
- le web sémantique (RDF/XML)

*Ou encore pour*

- des images vectorielles (SVG)
- des formules mathématiques (MathML)
- des notations musicales (MEI)

*etc.*

Pour une liste, un peu datée (2005) : http://xml.coverpages.org/xmlApplications.html

```
<TEI>
   <teiHeader> [...]   </teiHeader>
   <text> <!-- début du corps du document TEI -->
      <body>
         <p>Et voici la partition de ladite chanson.</p>
          <notatedMusic>

            <mei>
             <meiHead> (...) </meiHead>
             <music>
                <body>
                   <mdiv>
                      <score>
                         (...) <!-- la partition, en MEI -->
                      </score>
                   </mdiv>
                </body>
             </music>
            </mei>

          </notatedMusic>
      </body>
   </text>
</TEI>
```



*Comment distinguer les deux vocabulaires ?*

# Les espaces de noms XML

- permettent d'utiliser des éléments et attributs issus de **plusieurs langages XML** dans un même document, sans les confondre
- chacun des éléments du document est assigné à un espace de nom particulier, déclaré par le biais d'un **URI** et matérialisé par un **préfixe** ;
- la déclaration d'espace de nom se fait à l'aide d'un pseudo-attribut xmlns qui doit se trouver au moins dans l'élément le plus haut faisant emploi du langage XML concerné (mais peut se trouver encore plus en amont, par exemple au niveau de l'élément racine du document)

```
<tei:TEI xmlns:tei="http://www.tei-c.org/ns/1.0">
   <tei:teiHeader> (...)   </tei:teiHeader>
   <tei:text> <!-- début du corps du document TEI -->
      <tei:body>
         <tei:p>Et voici la partition de ladite chanson.</tei:p>
          <tei:notatedMusic>
           <mei:mei xmlns:mei="http://www.music-
encoding.org/ns/mei">
             <mei:meiHead> (...) </mei:meiHead>
             <mei:music>
                <mei:body>
                   <mei:mdiv>
                      <mei:score>
                         (...) <!-- la partition, en MEI -->
                      </mei:score>
                   </mei:mdiv>
                </mei:body>
             </mei:music>
           </mei:mei>
       </tei:notatedMusic>
    </tei:body>
 </tei:text>
</tei:TEI>
```

# N.B.

Il est préférable, pour plus de lisibilité, de définir tous les espaces de noms à la **racine** du document

```
<TEI xmlns:tei="http://www.tei-c.org/ns/1.0"
xmlns:mei="http://www.music-encoding.org/ns/mei">
   <tei:teiHeader> (...)
```

Il est possible également de définir l'un des espaces de nom comme **par défaut** ; tous les éléments non préfixés lui seront rattachés

```
<TEI xmlns="http://www.tei-c.org/ns/1.0"
xmlns:mei="http://www.music-encoding.org/ns/mei">
   <teiHeader> (...)
```

# QName

## tei:teiHeader

| | |
|---|---|
| préfixe | tei |
| séparateur | : |
| nom local | teiHeader |

tei:teiHeader pourrait se lire
{http://www.tei-c.org/ns/1.0}teiHeader

Voir la spécification du W3C (Namespaces in XML 1.0 (Third Edition), en ligne : http://www.w3.org/TR/REC-xml-names/#NT-QName)

# Un travail de modélisation consubstantiel de XML

« **Modélisation** : Opération par laquelle on établit le modèle d'un système complexe, afin d'étudier plus commodément et de mesurer les effets sur ce système des variations de tel ou tel de ses éléments composants » (J. Giraud, P. Pamart (Pierre), J. Riverain, *Les nouveaux mots « dans le vent »*, Paris, France, 1974).

# Quelques écueils à éviter

- XML n'est pas une fin en soi / XML pour l'amour de l'art
- « Don't invent XML languages »
- Difficulté à définir de (bonnes) pratiques, et à les uniformiser
- Éviter la « balisologie » et surtout la « balisomanie » => plutôt, songer, lors de l'élaboration de l'encodage, en termes d'utilisation finale et d'exploitation des données

# Récapitulatif des définitions

1) XML est un *eXtensible Markup Language*, c'est-à-dire un métalangage utilisant un balisage sémantique et structuré des contenus ;
2) XML n'est pas caractérisé par un vocabulaire donné, il définit simplement des règles sur ce que doit être un document bien formé, ainsi qu'un document valide, et permet de créer différents langages ou applications
3) XML est un sous-ensemble de SGML, créé à l'origine pour les documents textuels (« narratifs ») mais qui est appliqué à toutes sortes de données textuelles
4) XML est un outil souple et puissant, et fournit un environnement de travail riche, mais il n'est pas la réponse à tous les besoins, ni la solution à tous les problèmes.

# Bibliographie indicative

*En complément des références données au cours de la présentation.*

## Introduction à XML

HAROLD (Elliotte Rusty) & MEANS (W. Scott), *XML en concentré: manuel de référence*, 3e éd., Paris, 2005, part. les chap. I à IV.

TEI CONSORTIUM, « v. A Gentle Introduction to XML », dans EID., *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, 2014, en ligne : <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/SG.html> ( consulté en nov. 2014).

VAN DEN BRANDEN (Ron), TERRAS (Melissa) & VANHOUTTE(Edward), *TEI by Example*, <http://www.teibyexample.org> (consulté en nov. 2013).

WORLD WIDE WEB CONSORTIUM, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation, 26 nov. 2008, en ligne : http://www.w3.org/TR/REC-xml/