

Introduction à XML

*

Objectifs

- Connaître les principales caractéristiques de ce formalisme
- Être capable de créer et d'exploiter des documents en XML

*

*Ce document d'enseignement est diffusé librement, pour usage individuel.
Il est librement téléchargeable sur le site de l'auteur *.*

Michel Cartereau - Septembre 2016

PRÉSENTATION GÉNÉRALE

PRINCIPALES CARACTÉRISTIQUES

UN EXEMPLE SIMPLE D'INTRODUCTION

STRUCTURATION

BALISAGE

DOCUMENT EN XML

VISUALISATION DU DOCUMENT

QU'EST-CE QUE XML ?

XML : *EXTENSIBLE MARKUP LANGUAGE* ¹



- UNE NOTATION

- Définition formelle

- un métalangage de balisage (descendant du langage SGML ²), conçu vers 1997 afin de faciliter l'échange de données via le Web

- Codage de données et aussi de documents

- structuration par balisage d'éléments de contenu

- Objectifs de simplicité, généralité et de facilité d'utilisation

- format universel de texte brut, auto-descriptif, normalisé et ouvert

- Adaptation aux traitements automatiques

- disponibilité d'outils génériques d'analyse et de transformation

- LA BASE DE NOMBREUSES APPLICATIONS

- Une multitude de formats dérivés

- pages Web (XHTML), documents (OpenDocument ³, OpenXML ⁴), images vectorielles (SVG), expressions mathématiques (MathML), présentations multimedia (SMIL), syndication (Atom, RSS), Web sémantique (RDF, SKOS, OWL), etc.

- Une grande famille de technologies

- manipulation des éléments (DOM, XPath), transformation (XSLT), mise en forme (CSS, XSLT-FO), validation (DTD, Schéma XML), interrogation (XQuery), etc.

NORMALISATION PAR LE W3C : [HTTP://WWW.W3.ORG/STANDARDS/XML/](http://www.w3.org/standards/xml/)
PRÉSENTATION ICI DE LA VERSION 1.0

¹ XML : langage de balisage extensible

² SGML (*standard generalized markup language*) : langage général de balisage de document normalisé en 1986, adapté à l'édition d'ouvrages ou la production de grands rapports.

³ OpenDocument : format ouvert de document développé par Sun puis le consortium Oasis, utilisé dans la suite bureautique OpenOffice, normalisé en 2006 (ISO/IEC 26300:2006).

⁴ OpenXML : format concurrent de OpenDocument développé par Microsoft, utilisé dans sa suite bureautique Office 2007, normalisé en 2008 (ISO/IEC 29500:2008).

EXEMPLE DE DONNÉES : BIBLIOGRAPHIE

EXEMPLE SIMPLIFIÉ



- CARACTÉRISTIQUES DES DONNÉES

- Structuration des données

Bibliographie constituée d'une liste de références

Référence identifiée par un libellé et un type (livre ou site)

Livre caractérisé par son titre, le nom - et le prénom facultatif - de son(es) auteur(s), et l'année de sa publication

Site caractérisé par son titre et son adresse (URL)

- Exemple avec 3 références (2 livres et 1 site) :

[Brillant 2007]

Brillant Alexandre. **XML cours et exercices**. 2007.

[North Hermans 2000]

North Simon, Hermans Paul. **Teach Yourself XML in 21 Days**. 2000.

[W3C XML]

Normalisation de XML par le W3C. En ligne à :
<http://www.w3.org/standards/xml/>.

- OBJECTIFS

- Enregistrement dans un format simple et ouvert
fichier directement imprimable et contenu lisible

- Edition

création et mises à jour, avec mise en évidence de la structure

- Manipulations

extraction de données basée sur leur structure

production automatisée d'une liste sous forme d'une page Web

UTILISATION A PRIORI D'OUTILS FACILEMENT DISPONIBLES

STRUCTURATION EN XML

ORGANISATION DES DONNÉES

• PRINCIPES DE CONSTRUCTION

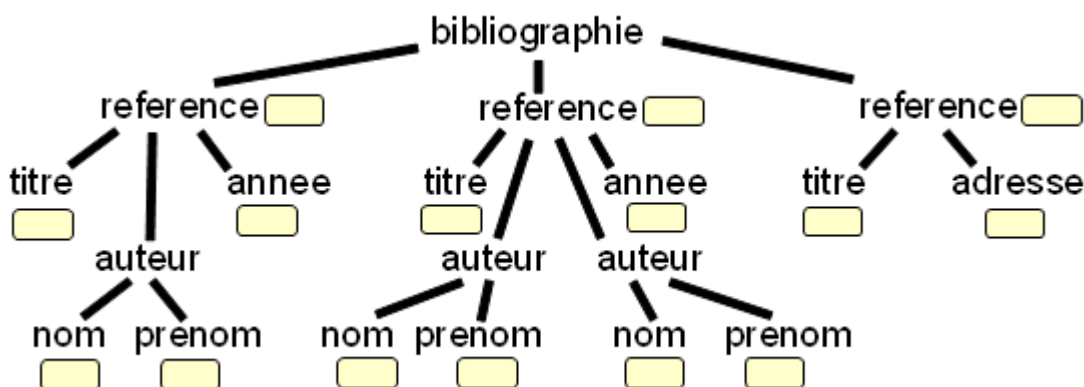
- Structuration en éléments de contenu
organisation hiérarchique sous la forme d'une arborescence
contenu constitué par les données sous forme textuelle
- Types d'éléments
l'élément principal représentant l'ensemble des données
éléments terminaux correspondant aux données élémentaires
éléments intermédiaires regroupant un ou plusieurs sous-éléments (avec aussi éventuellement des données incluses)

• REPRÉSENTATION EN XML

- Structuration arborescente en éléments de contenu
une balise associée à chaque élément à partir de l'élément principal
- Données associées aux balises
soit comme le contenu inséré dans la balise
soit comme le ou l'un des attributs de la balise ¹

• EXEMPLE

ici, le libellé et le type d'une référence sont associées à sa balise



¹ Le choix de placer la donnée en contenu ou en attribut dépend des usages ; généralement, l'attribut sert à indiquer une métadonnée (exemple : le type de référence, un livre ou un site).

BALISAGE EN XML

• CARACTÉRISTIQUES DU BALISAGE

- Balisage avec contenu :
contenu placé entre une balise ouvrante et une balise fermante :
<nom¹ d'élément attributs optionnels > contenu <nom d'élément />
- Balisage sans contenu
notation en une balise unique : *<nom d'élément attributs optionnels />*
- Attribut de balise
notation : *nom¹ d'attribut="valeur" ou nom d'attribut='valeur'²*

• EXEMPLE

```
<bibliographie>
  <reference id-ref="Brillant 2007" type-ref="livre">
    <titre>XML cours et exercices</titre>
    <auteur>
      <nom>Brillant</nom> <prenom>Alexandre</prenom>
    </auteur>
    <annee>2007</annee>
  </reference>
  <reference id-ref="North Hermans 2000" type-ref="livre">
    <titre>Teach Yourself XML in 21 Days</titre>
    <auteur>
      <nom>North</nom> <prenom>Simon</prenom>
    </auteur>
    <auteur>
      <nom>Hermans</nom> <prenom>Paul</prenom>
    </auteur>
    <annee>2000</annee>
  </reference>
  <reference id-ref="W3C XML" type-ref="site">
    <titre>Normalisation de XML par le W3C</titre>
    <adresse>http://www.w3.org/standards/xml/</adresse>
  </reference>
</bibliographie>
```

¹ Un nom d'élément ou d'attribut se compose de lettres, chiffres ou de l'un des symboles « : » ou « - » ou « _ » ou « . » ; il doit débuter par une lettre ou bien l'un des symboles « : » ou « _ ».

² Réécriture du symbole délimiteur si réutilisé dans le texte de *valeur* ; guillemet (") : " ou " ; - apostrophe (') : ' ou '

DOCUMENT EN XML

VERSION MINIMALE DE DOCUMENT

● SYNTAXE GLOBALE

- Document constitué en 1^{ère} ligne d'un prologue suivi de l'élément principal
prologue conseillé : `<?xml version="1.0" encoding="codage" />`¹
déclaration de la version d'XML utilisée ("1.0" ici) et du codage
appliqué au texte (généralement UTF-8 ou ISO-8859-1)
- Commentaires optionnels
notation : `<!-- texte libre (sans la notation « -- ») -->`

● EXEMPLE

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- bibliographie.xml - Michel Cartereau - 12 XII 2009 -->
<bibliographie>
  <reference id-ref="Brillant 2007" type-ref="livre">
    <titre>XML cours et exercices</titre>
    <auteur>
      <nom>Brillant</nom>
      <prenom>Alexandre</prenom>
    </auteur>
    <annee>2007</annee>
  </reference>
  <reference id-ref="North Hermans 2000" type-ref="livre">
    <titre>Teach Yourself XML in 21 Days</titre>
    <auteur>
      <nom>North</nom>
      <prenom>Simon</prenom>
    </auteur>
    <auteur>
      <nom>Hermans</nom>
      <prenom>Paul</prenom>
    </auteur>
    <annee>2000</annee>
  </reference>
  <reference id-ref="W3C XML" type-ref="site">
    <titre>Normalisation de XML par le W3C</titre>
    <adresse>http://www.w3.org/standards/xml/</adresse>
  </reference>
</bibliographie>
```

ENREGISTREMENT DANS UN FICHIER AU SUFFIXE .xml

¹ Cette balise doit figurer tout au début du document, sans aucun blanc avant (son absence est tolérée).
Quand le codage n'est pas indiqué, c'est UTF-8 (Unicode sur 8 bits) qui s'applique.

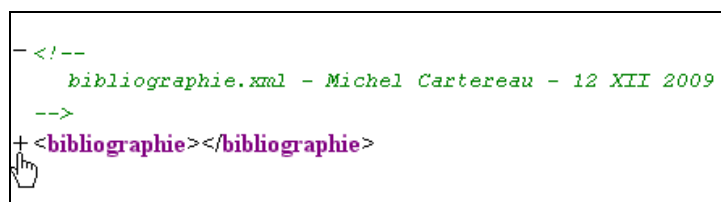
VISUALISATION DIRECTE DANS UN NAVIGATEUR

FORMAT XML RECONNU PAR LES NAVIGATEURS



■ Mise en forme simple du texte

affichage de l'arborescence des éléments avec indentation automatique,
et réduction possible dans certains navigateurs ¹

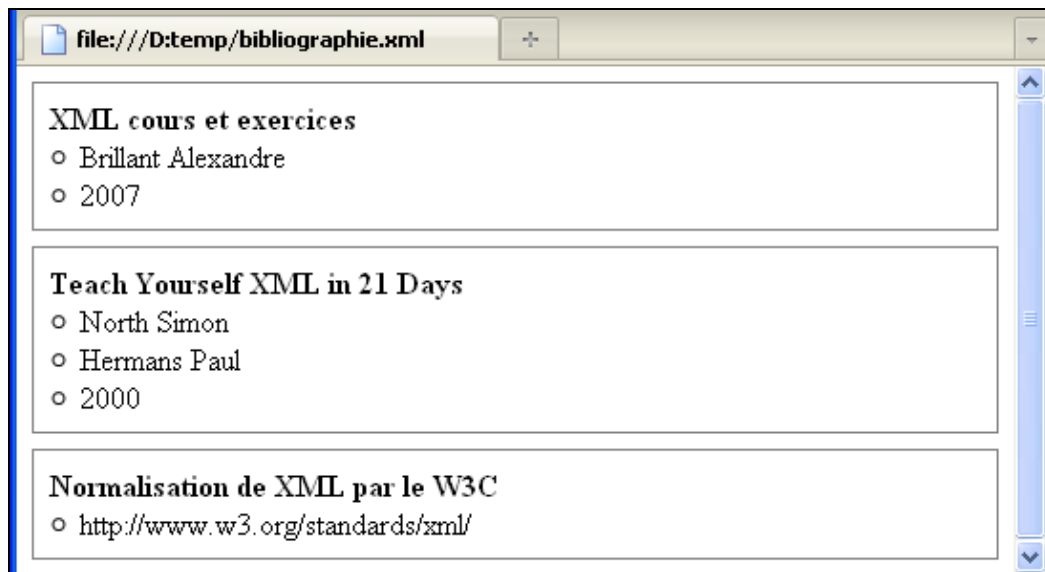


exemple de réduction d'un élément dans Firefox

¹ Cas de Mozilla Firefox, Google Chrome et Microsoft Internet explorer.

VISUALISATION AVEC UNE FEUILLE DE STYLE

AMÉLIORATION DE LA VISALISATION DU DOCUMENT



- Association possible d'une feuille de style (en CSS)

balise spécifique ¹ placée dans le prologue (généralement en fin) :

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

- Exemple

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- bibliographie.xml - Michel Cartereau - 12 XII 2009 -->
<?xml-stylesheet type="text/css" href="bibliographie.css"?>
```

avec la feuille de style suivante :

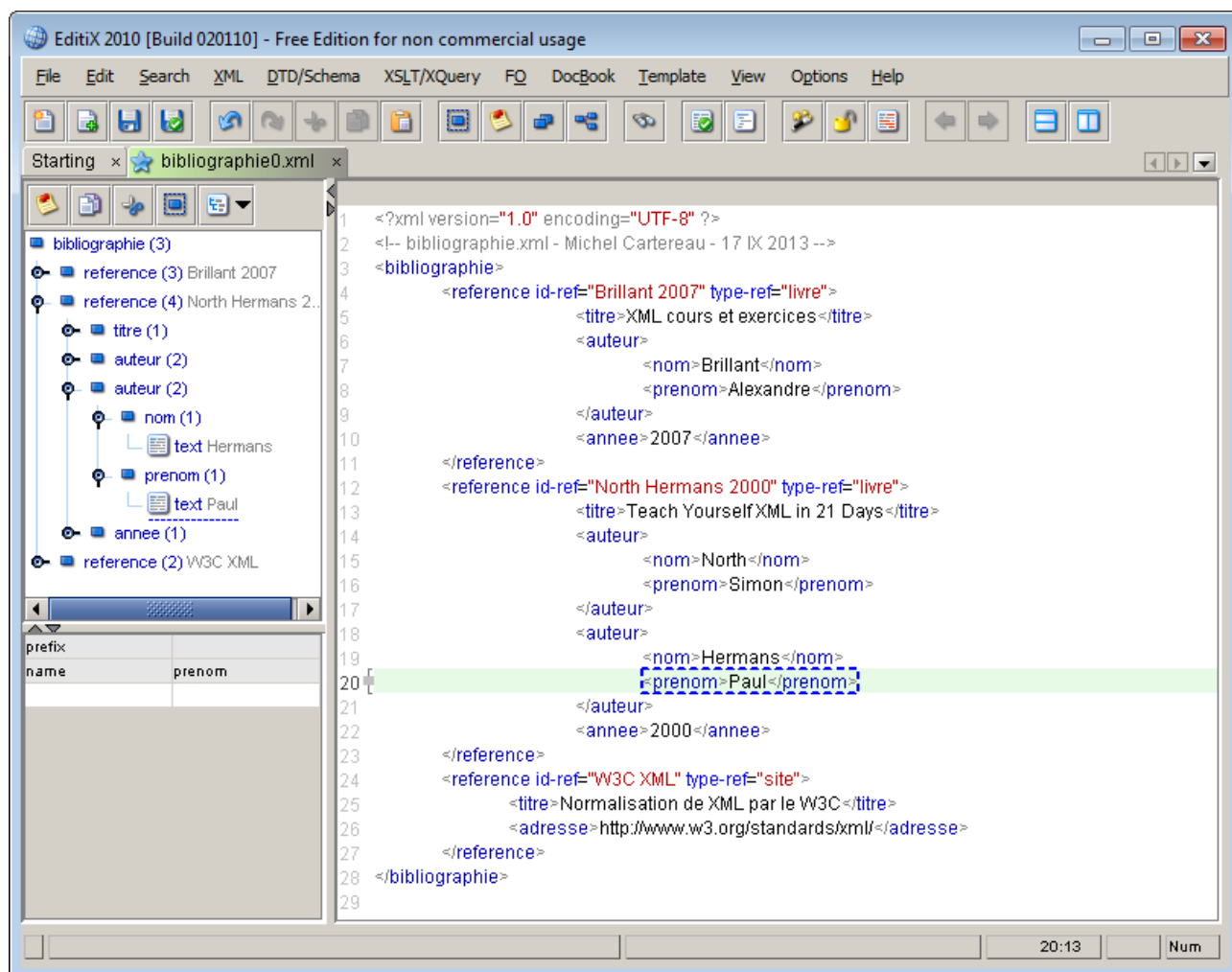
```
/* bibliographie.css - Michel Cartereau 12 XII 2009 */
reference {
    display: block;
    margin: 0.5em;
    border: thin solid gray;
    padding: 0.5em; }
titre {
    font-weight: bold;
    display: block; }
auteur, annee, adresse {
    display: list-item;
    list-style: circle;
    margin-left: 1em; }
```

APPLICATION DE STYLES EN CSS AUX ÉLÉMENTS DU DOCUMENT EN XML

¹ Balise dite de traitement (*processing instruction*) destinée au navigateur.

EXEMPLE D'ÉDITEUR SPÉCIALISÉ : EDITIX

CAS DE LA VERSION 2010 D'EDITIX POUR WINDOWS ¹



- Aide à la saisie des données
indication de l'arborescence des éléments
- Un outil plutôt complet
nombreuses manipulations possibles

AUTRES OUTILS GRATUITS : JAXE ², NOTEPAD++ ³

¹ EditiX : version d'essai gratuite (30 jours), pour Windows, Mac OS X et Linux, en <http://www.editix.com/>

² Jaxe : logiciel libre et gratuit, en java, pour Windows, Mac OS X et Linux, en <http://jaxe.sourceforge.net/>

³ Notepad++ : logiciel libre et gratuit pour Windows, en <http://notepad-plus-plus.org/>, à compléter avec l'extension XMLTools

EXERCICES SUR LA STRUCTURATION EN XML

1. Compléter les données en ajoutant le nom de l'éditeur pour les livres et la date de consultation pour un site.
2. On considère la référence d'un article publié dans une revue avec le ou les auteurs, le titre de l'article, le nom de la revue, l'année de publication, le numéro du volume si existant, celui du fascicule et les numéros de pages (intervalle) ; proposer une structuration en XML.
3. Que faut-il modifier dans le document en XML afin de pouvoir visualiser le libellé d'une référence ? (exemple : `Brillant 2007` pour la 1^{ère})
4. La donnée du type de référence est-elle nécessaire ?
5. Modifier la feuille de style de l'exemple afin de visualiser l'année dans un paragraphe et en taille réduite.
6. Proposer la représentation en XML d'un dossier de fichiers

TYPE DE DOCUMENT (DTD)

GÉNÉRALITÉS

PRÉSENTATION DU FORMALISME DE DTD

GÉNÉRALITÉS SUR LA VÉRIFICATION

LES TRAITEMENTS AUTOMATIQUES SONT CONÇUS
POUR S'APPLIQUER À DES DOCUMENTS CORRECTS



- DEUX NIVEAUX DE VÉRIFICATION
 - Premier niveau : bonne écriture en XML
respect des règles d'écriture des balises et du contenu
règles définies par la norme du W3C
 - Second niveau : cohérence de la structuration
types de données et relations de dépendances entre les éléments
nécessité d'exprimer ces contraintes dans un formalisme spécifique
- BONNE ÉCRITURE EN XML : DOCUMENT « BIEN ÉCRIT » OU « BIEN FORMULÉ » ¹
 - Principales règles d'écriture
un élément principal unique
pas de recouvrement des balises
même nom dans les balises d'ouverture et fermeture d'un élément
(et écrits dans la même casse ² de caractères)
symbole « < » interdit dans le texte en dehors de la notation de balise
(sauf dans une section spéciale dite `CDATA` - cf. page 16)
 - Grammaire définie pour XML
notation de la norme XML 1.0 ³ avec le métalangage EBNF ⁴
- COHÉRENCE DE LA STRUCTURATION : DOCUMENT « VALIDE »
 - Plusieurs formalismes pour exprimer la structuration ⁵ (« vocabulaires »)
DTD initialement puis Schéma XML du W3C (et aussi Relax NG ⁶)
 - Présentation ici du formalisme DTD (*document type definition*)
définition simple mais limitée des contraintes (inspirée de SGML)
existence de nombreux outils pour effectuer le travail de validation

DESCRIPTIONS FORMELLES À L'AIDE DE MÉTALANGAGES

¹ En anglais : *well-formed document*.

² Par convention, les noms sont écrits en lettres minuscules (bas de casse).

³ Voir la présentation complète en annexe.

⁴ EBNF (*extended Backus-Naur form*) : ancien métalangage adapté par le W3C pour ses besoins.

⁵ Cela correspond à un « type de document » ou un « schéma de structuration »

⁶ Relax NG (*regular language for XML next generation*) : défini par l'OASIS et l'ISO.

MODÈLE EN DTD (*DOCUMENT TYPE DEFINITION*)

DÉFINITION FORMELLE D'UN TYPE DE DOCUMENT

- CARACTÉRISTIQUES GÉNÉRALES

- Une liste de définitions
définition par une balise soit d'élément, soit d'attribut d'élément
- La définition d'un élément : `<!ELEMENT >`
indication du nom et de la nature du contenu (éléments ou valeurs)
- La définition d'attribut(s) d'élément : `<!ATTLIST >`
indication de l'élément en référence, puis du nom, de la valeur et du caractère obligatoire ou non de la présence de l'attribut

- EXEMPLES INTRODUCTIFS

- Élément de bibliographie
une liste de références, éventuellement vide
`<!ELEMENT bibliographie (reference)* >`
- Élément de référence
un titre puis aucun, un ou plusieurs auteurs, puis une adresse facultative, puis une année de publication facultative
`<!ELEMENT reference (titre, auteur*, adresse?, annee?) >`
- Élément de titre
une valeur notée textuellement
`<!ELEMENT titre (#PCDATA)>`
- Élément d'auteur
un nom, puis un prénom facultatif
`<!ELEMENT auteur (nom, prenom?) >`
- Attribut d'identification d'une référence
une valeur notée textuellement et obligatoire
`<!ATTLIST reference id-ref CDATA #REQUIRED>`
- Attribut du type d'une référence
une valeur au choix parmi « livre » ou « site », et obligatoire
`<!ATTLIST reference type-ref (livre | site) #REQUIRED>`

NOTATION DE DTD (*DOCUMENT TYPE DEFINITION*)

ECriture EN SGML, PRÉSENTATION ICI SIMPLIFIÉE ¹

● CARACTÉRISTIQUES GÉNÉRALES

- Définition d'un élément : `<!ELEMENT nom contenu >`
soit un contenu vide : `EMPTY`
soit un contenu quelconque sans structure : `ANY` ²
soit un simple texte : `(#PCDATA)` ³
soit un groupe d'un ou plusieurs sous-éléments : `(groupe)`
- Définition d'un attribut d'élément : `<!ATTLIST élément attribut >`
ou de plusieurs attributs à la fois : `<!ATTLIST élément attribut1 attribut2 ... >`
par un triplet d'informations : `nom type d'attribut valeur d'attribut`

● CARACTÉRISTIQUES DÉTAILLÉES

- Nom d'élément ou d'attribut
composé de lettres ⁴, chiffres, « `_` », « `-` », « `.` », « `:` »
débuté par une lettre, « `:` » ou « `_` »
- Groupe d'un ou plusieurs sous-éléments
soit une séquence de termes : `(terme1, terme2 ...)`
soit une alternative de termes : `(terme1 | terme2 ...)`
où un terme est : un nom d'élément, une séquence, ou une alternative
et pour chaque terme, séquence ou alternative, la possibilité d'indiquer ⁵ :
soit une présence optionnelle (0 ou 1 fois) : `?`
soit une présence 0, 1 ou plusieurs fois : `*`
soit une présence 1 ou plusieurs fois : `+`
- Type d'attribut
soit du texte : `CDATA`
soit une liste de valeurs possibles : `(valeur1 | valeur2 | ...)`
soit un type spécial : `ID` (identifiant ⁶) `IDREF` (référence d'un identifiant)
`IDREFS` (plusieurs ⁷ références) `NMTOKEN` (mot ⁸) `NMTOKENS` (plusieurs mots)
- Valeur d'attribut
soit obligatoire : `#REQUIRED`
soit facultative : `#IMPLIED`
soit fixée à une valeur constante : `#FIXED "valeur"` ou `#FIXED 'valeur'`
soit avec une valeur par défaut en cas d'absence : `"valeur"` ou `'valeur'`

¹ Voir la grammaire d'écriture en annexe.

² Le type `ANY` est déconseillé car les éventuels sous-éléments contenus dans le texte sont reconnus mais il n'y a alors pas de contrôle sur leur validité (élément `ANY` sans structure interne)

³ `PCDATA` signifie « *parsable character data* » (données de texte reconnaissables par un analyseur).

⁴ Il est sage de se limiter aux lettres de l'alphabet anglais, en évitant notamment les accents.

⁵ Le symbole ajouté doit être collé au texte précédent, c'est-à-dire sans blanc avant.

⁶ Un identifiant s'écrit comme un nom d'élément ; sa valeur doit être unique dans le document.

⁷ Plusieurs références (ou mots) doivent être séparées par au-moins un espace.

⁸ Un mot correspond à du texte sans blanc (*whitespace*) à l'intérieur : espace, tabulation ou retour-à-la-ligne.

NOTATION D'UNE DTD : COMPLÉMENTS

- Ecriture d'une notation

aération optionnelle à l'intérieur et entre les balises avec un ou plusieurs blancs (espace, tabulation ou retour-à-la-ligne)

parenthèse ouvrante (« (») toujours précédée d'au-moins un blanc

- Commentaire

placement libre, à l'extérieur d'une définition

notation : `<!-- texte libre (mais sans la notation « -- ») -->`

exemple : `<!-- bibliographie.dtd - Michel Cartereau -->`

- Entité

mécanisme d'abréviation avec association d'un nom ¹ à un texte

cas de texte pour XML : `<!ENTITY nom "texte">` ou `<!ENTITY nom 'texte'>`

et réutilisation par la notation : `&nom;`

exemple : `<!ENTITY APT "AgroParisTech" >`

avec en XML : `<auteur>&APT;</auteur>`

cas d'entité prédéfinie pour un symbole utilisé dans les notations :

`&` (« & ») `<` (« < ») `>` (« > ») `"` (« " ») `'` (« ' »)

exemple (XML) : `<comparaison>taille & 100</comparaison>`

cas d'entité interne à un document de DTD (et non dans XML) :

`<!ENTITY % nom "texte">` ou `<!ENTITY % nom 'texte'>`

réutilisation par la notation : `%nom;`

exemple : `<entity % couleur "(rouge|vert|bleu)">`

et `<!ATTLIST etiquette couleur-etiq %couleur; #IMPLIED>`

cas d'entité définie dans une autre DTD : `<!ENTITY nom SYSTEM "URL">`

exemple : `<entity adresse SYSTEM "courriel.dtd">`

- Symbole particulier

noté par le code décimal ou hexadécimal dans l'alphabet de référence,

soit `&#code;` soit `odeh;`

exemples (UTF-8) : `	` (tabulation) `©` (©)

- Notation de texte

valeur d'attribut, `CDATA` (*character data*)

tout symbole sauf « & » et « < »

valeur d'un élément, `PCDATA` (*parsed character data*)

référence d'entité (« &...; ») possible mais pas de balise (« <...> »)

exemple : `©` Carotte DU JARDIN, 2000

peut contenir une section spéciale `CDATA` : texte sans aucune contrainte (balises permises) mais délimité par `<![CDATA[` au début, `]]>` à la fin

exemple : voici un filet en XHTML : `<![CDATA[<hr />]]>`

¹ Le nom s'écrit comme un nom d'élément.

INDICATION D'UNE DTD POUR LE DOCUMENT

INDICATION FACULTATIVE MAIS CONSEILLÉE

● CAS DE DTD DANS UN FICHIER EXTERNE AU DOCUMENT EN XML

- Balise spécifique de définition du type du document en XML
indication de l'élément principal et du fichier de la DTD :

```
<!DOCTYPE élément principal SYSTEM "adresse du fichier">
```

placement dans le prologue, après la déclaration de la version

- Exemple

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

```
<!DOCTYPE bibliographie SYSTEM "bibliographie.dtd">
```

avec le texte de la DTD dans le fichier `bibliographie.dtd` :

```
<!-- bibliographie.dtd - Michel Cartereau - 16 IX 2013 -->
<!ELEMENT bibliographie (reference)*>
<!ELEMENT reference (titre, auteur*, adresse?, annee?)>
<!ATTLIST reference id-ref CDATA #REQUIRED>
<!ATTLIST reference type-ref (livre | site) #REQUIRED>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT auteur (nom, prenom?) >
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT adresse (#PCDATA)>
<!ELEMENT annee (#PCDATA)>
```

● CAS DE DTD INTERNE AU DOCUMENT EN XML

- Autre forme de définition du type du document en XML
indication de l'élément principal et des déclarations de la DTD :

```
<!DOCTYPE élément principal [ déclarations ] >
```

- Exemple

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

```
<!DOCTYPE bibliographie [
```

```
<!ELEMENT bibliographie (reference)*>
```

```
...
```

```
<!ELEMENT annee (#PCDATA)>
```

```
] >
```

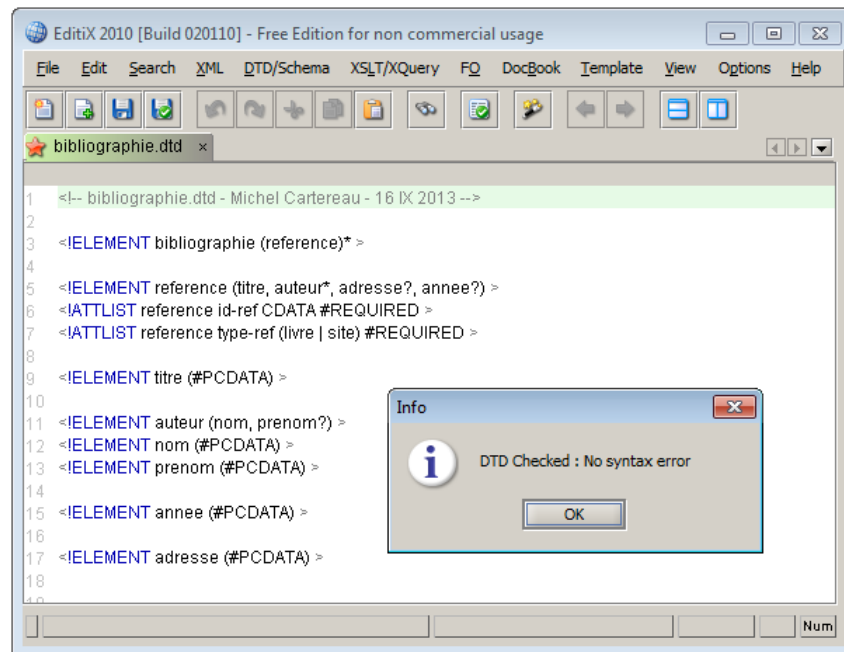
POSSIBILITÉS DE CUMULER LES DEUX FORMES (EXTERNE ET INTERNE ¹⁾)
MAIS DÉFINITION EXTERNE CONSEILLÉE CAR RÉUTILISABLE

¹ Dans le cas de la présence simultanée de déclarations externes et internes, priorité est donnée aux déclarations internes au document.

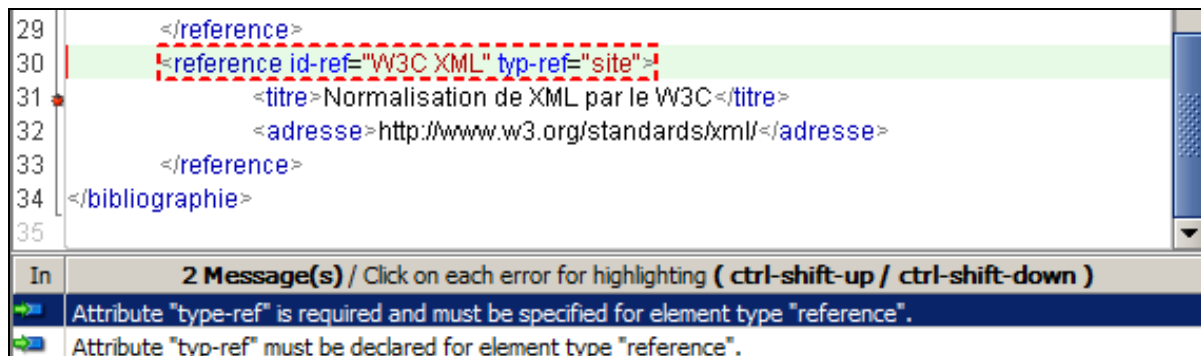
MANIPULATION DE DTD AVEC EDITIX

CAS DE LA VERSION 2010 D'EDITIX POUR WINDOWS

- Vérification de la bonne écriture d'une DTD
commande DTD/SCHEMA, CHECK THIS DTD
(si besoin, fixer le type de fichier par : VIEW, WINDOWS, FILE INFO)

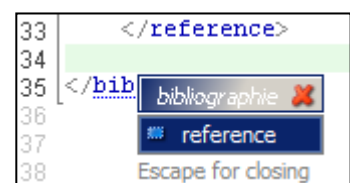


- Vérification de la bonne écriture et validation d'un document en XML
commande XML CHECK THIS DOCUMENT



cas d'une erreur avec sa localisation dans le texte

- Assistance à la saisie dans le document en XML
proposition automatique d'un élément selon la DTD
du document dès la frappe de « < »



EXERCICES SUR LES DTD

1. Indiquer si la DTD de l'exemple de la bibliographie autorise :
 - a) un livre sans auteur
 - b) un site sans identification de référence
 - c) un site sans adresse
 - d) l'attribution de plusieurs auteurs à un site
 - e) un auteur avec un nom écrit en deux parties

2. Modifier la DTD de l'exemple de la bibliographie afin de :
 - a) ajouter une référence de publication dans une revue (un article)
 - b) permettre l'indication de la langue utilisée pour rédiger un livre
 - c) imposer la présence d'une adresse dans le cas d'un site

3. Concevoir une DTD pour le cas d'une adresse postale

4. Proposer une DTD pour la représentation d'un dossier de fichiers

ESPACES DE NOMMAGE

PRINCIPES

MISE EN OEUVRE

EXEMPLES



MÉLANGE DE SCHÉMAS DE DÉFINITION DANS UN DOCUMENT EN XML

● EXEMPLE

- Ajout à la bibliographie d'adresses de bibliothèque
bibliothèque(s) où le document est disponible à la consultation
liste des bibliothèques disponible dans une autre application de XML ¹

```
<!-- liste-bibli.dtd -
Michel Cartereau - 18 IX 2013 -->
<!ELEMENT liste-bibli (bibliotheque)+ >
<!ELEMENT bibliotheque (nom, adresse)>
<!ATTLIST bibliotheque id-bibli
ID #REQUIRED>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT adresse (#PCDATA)>

<?xml version="1.0" encoding="UTF-8"?>
<!-- liste-bibli.xml -
Michel Cartereau - 18 IX 2013 -->
<!DOCTYPE liste-bibli
SYSTEM "liste-bibli.dtd">
<liste-bibli>
  <bibliotheque id-bibli="bib01">
    <nom>AgroParisTech</nom>
    <adresse>16 rue Claude-Bernard -
      75005 Paris</adresse>
  </bibliotheque>
  <bibliotheque id-bibli="bib02">
    <nom>UPMC</nom>
    <adresse>4 place Jussieu -
      75005 Paris</adresse>
  </bibliotheque>
</liste-bibli>
```

- Nouveau document fusionnant bibliographie et liste des bibliothèques
ajout de l'élément `bibliotheque` comme sous-élément de `reference`
[...]

```
<reference id-ref="Brillant 2007" type-ref="livre">
  <titre>XML cours et exercices</titre>
  <auteur>
    <nom>Brillant</nom> <prenom>Alexandre</prenom>
  </auteur>
  <annee>2007</annee>
  <bibliotheque id-bibli="bib02">
    <nom>UPMC</nom>
    <adresse>4 place Jussieu - 75005 Paris</adresse>
  </bibliotheque>
</reference>
```

[...]

confusion pour l'élément `nom` : auteur ou bibliothèque ?
(idem pour l'élément `adresse`)

SITUATION D'AMBIGUÏTÉ À ÉVITER

¹ Application de XML (ou « application XML ») : utilisation de XML dans un domaine précis, caractérisée par son schéma de structuration défini explicitement (DTD par exemple) ou non.

PRINCIPES DE L'ESPACE DE NOMMAGE

INDICATION DU CONTEXTE D'INTERPRÉTATION D'UN ÉLÉMENT OU D'UN ATTRIBUT

● OBJECTIFS

- Lever l'ambiguïté sur des noms identiques
association du contexte d'interprétation : l'« espace de nommage ¹ »
(correspond en fait à l'application de XML où le nom est utilisé)
- Faciliter le traitement avec des contextes d'interprétation multiples
identification unique des espaces de nommage utilisés

● MÉCANISMES

- Identification d'un espace de nommage par une URI ²
identification par URI en fait formelle sans analyse automatique ³
exemple : `http://purl.org/dc/` (Dublin core)
- Déclaration d'espace de nommage au niveau d'un élément
contexte utilisable par l'élément et toute sa descendance (pas au-delà)
avec a priori l'association d'une abréviation ⁴ à l'espace de nommage
notation par un attribut spécifique d'élément : `xmlns:préfixe=URI`
possibilité d'attribuer plusieurs espaces de nommages (préfixe distincts)
exemples : `<livre xmlns:dc="http://purl.org/dc/" >`
`<bibliographie xmlns:bib="mailto:auteur@ici.fr" >`
`<publication xmlns:dc="http://purl.org/dc/"`
`xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax" >`
- Indication de l'espace de nommage associé à un élément
notation avec l'abréviation en « préfixe » du nom : `préfixe:nom`
(nom d'élément ou d'attribut dit « qualifié », « *QNAME* » ou « complet »)
notation à répéter pour les attributs définis dans l'espace de nommage
exemples : `<dc:title>` `bib:id-bibli`
- Cas particulier d'un espace de nommage défini par défaut
contexte utilisé automatiquement en l'absence de préfixage des noms
notation sans utilisation d'un préfixe : `xmlns=URI`
exemple : `<html xmlns="http://www.w3.org/1999/xhtml">`
définition automatique dans le cas général d'un document en XML :
`xmlns="http://www.w3.org/XML/1998/namespace"`

ESPACES DE NOMMAGE DÉFINIS PAR LE W3C JUSTE APRÈS LA VERSION 1.0 DE XML
<http://www.w3.org/TR/xml-names/>

¹ Espace de nommage ou « *namespace* » en anglais.

² URI (*uniform resource indicator*) : identification unique d'une ressource sur le Web ; correspond de fait à une adresse (URL, *uniform resource locator*) sous la forme `http://`, `ftp://` etc.

³ L'URI d'espaces de nommage publics correspond généralement à une documentation.

⁴ Le choix de l'abréviation est libre ; elle est locale à l'élément en XML ; elle suit les mêmes règles que pour un nom d'élément mais sans deux-points (« : ») possible.

EXEMPLE DE LA BIBLIOGRAPHIE LOCALISÉE

ESPACE DE NOMMAGE ATTRIBUÉ À LA LISTE DE BIBLIOTHÈQUES

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- bibliographie-localisee.xml - Michel Cartereau 19 IX 2013 -->

<bibliographie xmlns:bib="mailto:auteur@ici.fr">
  <reference id-ref="Brillant 2007" type-ref="livre">
    <titre>XML cours et exercices</titre>
    <auteur>
      <nom>Brillant</nom>
      <prenom>Alexandre</prenom>
    </auteur>
    <annee>2007</annee>
    <bib:bibliotheque bib:id-bibli="bib02">
      <bib:nom>UPMC</bib:nom>
      <bib:adresse>4 Place Jussieu - 75005 Paris</bib:adresse>
    </bib:bibliotheque>
  </reference>
  [...]
</bibliographie>
```

ou dans une seconde version équivalente :

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- bibliographie-localisee.xml - Michel Cartereau 19 IX 2013 -->

<bibliographie>
  <reference id-ref="Brillant 2007" type-ref="livre">
    <titre>XML cours et exercices</titre>
    <auteur>
      <nom>Brillant</nom>
      <prenom>Alexandre</prenom>
    </auteur>
    <annee>2007</annee>
    <bib:bibliotheque bib:id-bibli="bib02"
      xmlns:bib="mailto:auteur@ici.fr">
      <bib:nom>UPMC</bib:nom>
      <bib:adresse>4 Place Jussieu - 75005 Paris</bib:adresse>
    </bib:bibliotheque>
  </reference>
  [...]
</bibliographie>
```

EXEMPLE D'UN FIL D'INFORMATION (RSS)

ACCÈS AUX ACTUALITÉS D'UN SITE (*REALLY SIMPLE SYNDICATION*) AVEC INSERTION DE MÉTADONNÉES DU *DUBLIN CORE*

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <channel xml:lang="fr">
    <title>AgroParisTech - Actualité</title>
    <link>http://www.agroparistech.fr/</link>
    <description>L'Institut des sciences et industries du vivant
      et de l'environnement, AgroParisTech, et son &#233;cole
      interne, l'ENGREF, forment des ing&#233;nieurs agronome, pour
      les industries alimentaires, l'environnement.</description>
    <language>fr</language>
    <generator>SPIP - www.spip.net</generator>
    <image>
      <title>AgroParisTech</title>
      <url>http://www.agroparistech.fr/local/siteon0-32.png</url>
      <link>http://www.agroparistech.fr/</link>
      <height>36</height>
      <width>144</width>
    </image>
    <item xml:lang="fr">
      <title>Conf&#233;rence annuelle de l'Institut
        Europ&#233;en des For&#234;ts (EFI) &#224; Nancy</title>
      <link>http://www.agroparistech.fr/+Conf+.html</link>
      <guid isPermaLink="true">
        http://www.agroparistech.fr/+Conf+.html</guid>
      <dc:date>2013-09-22T22:00:00Z</dc:date>
      <dc:format>text/html</dc:format>
      <dc:language>fr</dc:language>
      <description>
        <![CDATA[
          Cette ann&#233;e &#224; l'occasion des 20 ans de l'EFI
          (European Forest Institut), AgroParistech, l'Inra,
          l'Universit&#233; de Lorraine, et autres partenaires
          co-organisent une conf&#233;rence sur la for&#234;t qui se
          d&#233;roulera au Palais des Congr&#232;s de Nancy du 23 au
          27 septembre 2013. Au programme, un forum sur les (...)
        ]]>
      </description>
    </item>
  </channel>
</rss>
```


DTD ET ESPACE DE NOMMAGE

NOMS QUALIFIÉS À PRENDRE EN COMPTE DANS LE SCHÉMA DE STRUCTURATION

● CONTRAINTES

- Concept d'espace de nommage inconnu en DTD
pas de possibilité d'y déclarer un espace de nommage
- Obligation de déclarer les noms qualifiés d'élément et d'attribut
notation du nom avec son préfixe comme dans le document en XML

● EXEMPLE DE LA BIBLIOGRAPHIE LOCALISÉE

```
<!-- bibliographie-localisee.dtd - Michel Cartereau 19 IX 2013 -->

<!ELEMENT bibliographie (reference)* >
<!ATTLIST bibliographie xmlns:bib CDATA #REQUIRED>
<!ELEMENT reference (titre, auteur*, adresse?, annee?,
  bib:bibliotheque*) >
<!ATTLIST reference id-ref CDATA #REQUIRED >
<!ATTLIST reference type-ref (livre | site) #REQUIRED >
<!ELEMENT titre (#PCDATA) >
<!ELEMENT auteur (nom, prenom?) >
<!ELEMENT nom (#PCDATA) >
<!ELEMENT prenom (#PCDATA) >
<!ELEMENT annee (#PCDATA) >
<!ELEMENT adresse (#PCDATA) >

<!ELEMENT bib:bibliotheque (bib:nom, bib:adresse)>
<!ATTLIST bib:bibliotheque bib:id-bibli ID #REQUIRED>
<!ELEMENT bib:nom (#PCDATA)>
<!ELEMENT bib:adresse (#PCDATA)>
```

● ASTUCE

- Possibilité de fixer l'espace de nommage dans une déclaration en DTD
déclaration de l'attribut `xmlns` avec indication d'une valeur par défaut
exemple :

```
<!ATTLIST bibliographie xmlns:bib CDATA
  #FIXED "mailto:auteur@ici.fr">
```

EXERCICES SUR LES ESPACES DE NOMMAGE

ON CONSIDÈRE LE CAS DE LA BIBLIOGRAPHIE LOCALISÉE

1. Ajouter une bibliothèque à une autre référence
2. Est-ce que `<bib:bibliotheque bib:id-bibli="bib02">` est équivalent à `<bibliotheque bib:id-bibli="bib02">` ?
3. `<bib:bibliotheque id-bibli="bib02">` est-il valide ?
4. Indiquer la langue d'écriture du texte d'un livre ou d'un site en utilisant la métadonnée `language` de *Dublin core*
5. Modifier la DTD afin de prendre en compte la métadonnée introduite précédemment, avec une présence optionnelle
6. Retoucher la DTD afin d'attribuer automatiquement l'espace de nommage du *Dublin core* à tout document associée à cette DTD

SCHÉMA XML

GÉNÉRALITÉS

PRINCIPALES CARACTÉRISTIQUES

EXEMPLE

PRÉSENTATION DE SCHÉMA XML

DESCRIPTION AVANCÉE DE DOCUMENT, DÉFINIE PAR LE W3C EN 2001

● PRINCIPALES CARACTÉRISTIQUES

- Ecriture en XML
espace de nommage spécifique, préfixé par convention avec `xs:`
par convention, fichier au suffixe `.xsd`
- Spécification élaborée de types de donnée (contenus et attributs)
types prédéfinis : textuels, numériques, temporels etc.
domaines de valeurs définissables par des expressions rationnelles ¹
personnalisation de types par restriction etc.
- Prise en charge des espaces de nommage
espaces utilisés par le document et espace affecté au document
- Modularité
réutilisation de définitions de groupes d'éléments ou d'attributs
importation de sous-schémas

● STRUCTURE GÉNÉRALE

- Une liste de descriptions d'élément
balise `<xs:schema>` contenant une liste de balises `<xs:element>`
- Description du contenu d'un élément selon trois familles de types
types simples : valeurs élémentaires (texte, nombre, date, heure)
types complexes : sous-éléments (liste ordonnée ou non, au choix)
types mixtes : mélange de sous-éléments et de valeurs

- Modèle générique

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="élément1" type complexe </xs:element>
  <xs:element name="élément2" type="type simple">
    ...
</xs:schema>
```

PRÉSENTATION ICI SIMPLIFIÉE DE CE FORMALISME COMPLEXE ²

¹ Appelées aussi « expressions régulières » (*regular expression*), popularisées par le langage Perl

² Ouvrage de référence pour approfondir : Eric Van der Vlist, **XML Schéma**. O'Reilly, 2002, 400 p.

SCHÉMA XML : TYPES DE DONNÉES

PRÉSENTATION SIMPLIFIÉE

● TYPES ÉLÉMENTAIRES DE DONNÉES

▪ Textes

texte avec ses éventuels blancs conservés ¹ : `xs:string`

texte avec ses éventuels blancs normalisés ² : `xs:normalizedString`

texte avec ses éventuels blancs normalisés et compactés ³ : `xs:token`

▪ Nombres

nombre entier (signe possible : + ou -) : `xs:integer`

nombre réel simple ou double précision (IEEE) : `xs:float` ou `xs:double`

valeur logique : `xs:boolean` (valeurs `true` ou `1`, `false` ou `0`)

▪ Temps (notation selon norme ISO 8601 ⁴)

horodate : `xs:dateTime` - date : `xs:date` - heure : `xs:time`

intervalle de temps : `xs:duration`

▪ Valeurs binaires

notation en hexadécimal : `xs:hexBinary`

● TYPES DÉRIVÉS

▪ Principale technique : dérivation par restriction

restriction du domaine des valeurs possibles par définition de contraintes appliquées à certaines caractéristiques (« facettes ») d'un type existant

notation :

```
<xs:simpleType name="type dérivé">
  <xs:restriction base="type de base"> contraintes... </xs:restriction>
</xs:simpleType>
```

▪ Exemple

```
<xs:simpleType name="uneNote">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0">
    <xs:maxInclusive value="20">
  </xs:restriction>
</xs:simpleType>
```

NOTATION ASSEZ LOURDE

¹ Caractères blancs : espace (20_h), tabulation (09_h), saut-de-ligne (0A_h) et retour-à-la-ligne (0D_h)

² Normalisation de caractères blancs : tabulation, saut-de-ligne et retour-à-la-ligne remplacés par un espace

³ Compactage de caractères blancs : plusieurs espaces successifs remplacés par un espace unique, et élimination des espaces situés au début et à la fin.

⁴ Norme ISO 8601 : notation d'une horodate sous la forme `aaaa-mm-jjT hh:mm:ss` suivie en option par l'indication du fuseau horaire, soit `Z` (temps universel, TU), soit `±hh:mm` pour un décalage relatif au TU.

SCHÉMA XML : ATTRIBUT, ÉLÉMENT À CONTENU VIDE

PRÉSENTATION SIMPLIFIÉE

● ATTRIBUT

▪ Définition d'un attribut

indication du nom, type et du mode d'usage : obligatoire (*required*) ou non (*optional*, par défaut d'indication)

notation : `<xs:attribute name="nom" type="type" use="usage" />`

▪ Définition locale ou partagée

soit indication locale par placement dans la définition d'un élément

soit indication partagée entre plusieurs éléments par placement global au 1^{er} niveau du schéma puis réutilisation dans les éléments par la

notation : `<xs:attribute ref="nom" />`

▪ Exemples

```
<xs:attribute name="date-creation" type="xs:date" />
```

```
<xs:attribute name="note" type="uneNote" use="required" />
```

● ÉLÉMENT À CONTENU VIDE

▪ Élément avec uniquement des attributs

exemple : `<evaluation date-creation="2016-09-09" note="12" />`

indication de son nom, et du ou des attributs associés

▪ Notation

```
<xs:element name="nom">
```

```
  <xs:complexType>
```

```
    attributs...
```

```
  </xs:complexType>
```

```
</xs:element>
```

▪ Exemple

```
<xs:element name="evaluation">
```

```
  <xs:complexType>
```

```
    <xs:attribute ref="date-creation" />
```

```
    <xs:attribute ref="note" />
```

```
  <xs:complexType>
```

```
</xs:element>
```

SCHÉMA XML : ÉLÉMENT À CONTENU SIMPLE

PRÉSENTATION SIMPLIFIÉE

- ÉLÉMENT À CONTENU SIMPLE

- Élément avec une valeur mais sans sous-élément

exemple : `<module>publication sur l'Internet</module>`

indication du nom, du type et d'éventuels attributs associés

- Cas sans attributs

notation : `<xs:element name="nom" type="type" />`

ou `<xs:element name="nom d'élément" > définition du type </xs:element>`

- Cas avec attributs

indication des attributs dans la définition du type de la valeur ; notation :

```
<xs:element name="nom" >
  <xs:complexType>
    <xs:simpleContent>
      définition du type et des attributs
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

avec comme définition en cas de type simple:

```
<xs:extension base="type"> définition d'attribut... </xs:extension>
```

ou sinon :

```
<xs:restriction base="type de base">
  contrainte...
  définition d'attribut...
</xs:restriction>
```

- Exemples

```
<xs:element name="module" type="xs:string" />
```

```
<xs:element name="commentaire" >
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute ref="date-creation" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

SCHÉMA XML : ÉLÉMENT À CONTENU COMPLEXE, MIXTE

PRÉSENTATION SIMPLIFIÉE

● ÉLÉMENT À CONTENU COMPLEXE

▪ Élément se décomposant en sous-éléments

exemple : `<auteur> <nom>Chen</nom> <prenom>Shu</prenom> </auteur>`

indication du nom, des sous-éléments et d'éventuels attributs associés

▪ Regroupement des sous-éléments

soit une liste ordonnée (*sequence*) d'éléments (ou regroupements) :

`<xs:sequence> définition d'élément... </xs:sequence>`

soit un élément (ou regroupement) au-choix parmi (*choice*) plusieurs

`<xs:choice> définition d'élément... </xs:choice>`

soit une liste d'éléments (ou regroupements) en ordre quelconque (*any*)

`<xs:any> définition d'élément... </xs:any>`

▪ Cardinalités définies au niveau d'élément ou de regroupement

indication optionnelle à l'aide d'attributs du nombre d'occurrences :

minimum (*minOccurs*) et/ou maximum (*maxOccurs*), fixés à une seule occurrence par défaut d'indication, notation pour illimité (*unbounded*)

▪ Notation

`<xs:element name="nom" >`

`<xs:complexType> définition d'élément ou regroupement... </xs:complexType>`

`</xs:element>`

▪ Exemple

```
<xs:element name="auteur">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element name="nom" type="xs:string" />
```

```
<xs:element minOccurs="0" name="prenom" type="xs:string" />
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

● ÉLÉMENT À CONTENU MIXTE

▪ Élément se décomposant en sous-éléments et du texte

exemple : `<p>et voilà HTML 5</p>`

définition possible comme à contenu complexe, avec l'attribut

`mixed="true"`

▪ Notation

`<xs:complexType mixed="true" >`

EXEMPLE DE SCHEMA XML POUR LA BIBLIOGRAPHIE

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- bibliographie.xsd - Michel Cartereau - 8 IX 2016 -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bibliographie">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="reference"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="reference">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="titre"/>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="auteur"/>
        <xs:element minOccurs="0" ref="adresse"/>
        <xs:element minOccurs="0" ref="annee"/>
      </xs:sequence>
      <xs:attribute name="id-ref" use="required"/>
      <xs:attribute name="type-ref" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="livre"/>
            <xs:enumeration value="site"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
  <xs:element name="titre" type="xs:string"/>
  <xs:element name="auteur">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="nom"/>
        <xs:element minOccurs="0" ref="prenom"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="nom" type="xs:string"/>
  <xs:element name="prenom" type="xs:string"/>
  <xs:element name="annee" type="xs:string"/>
  <xs:element name="adresse" type="xs:string"/>
</xs:schema>
```

SÉLECTION DANS UN DOCUMENT AVEC XPATH

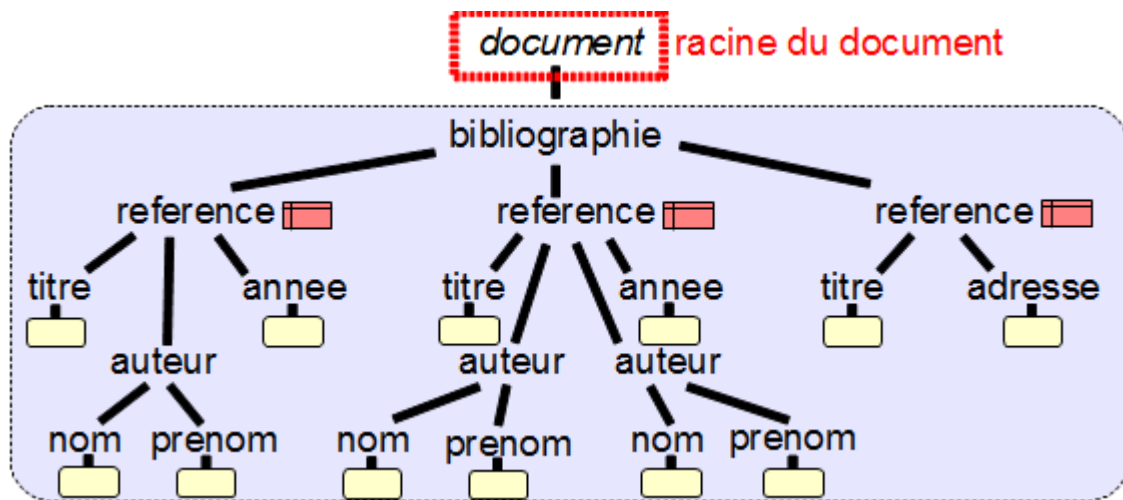
GÉNÉRALITÉS

CHEMIN DE SÉLECTION : AXE, CRITÈRE ET PRÉDICAT

ÉVALUATION D'UNE EXPRESSION

PRÉSENTATION DE XPATH (*XML PATH LANGUAGE*)

OPÉRATIONS BASÉES SUR UN REPÉRAGE D'ÉLÉMENTS
À SÉLECTIONNER DANS UN DOCUMENT EN XML



• PRINCIPES

- Document en XML représenté par une arborescence
la racine est le document, avec en fils unique l'élément principal de XML
(attention ! l'élément principal n'est pas considéré comme racine ici)
et sa sous-arborescence d'éléments et de nœuds de texte ¹
- Sélection dans la sous-arborescence de l'élément principal
repérage d'un ou plusieurs nœuds selon un chemin (*path*)
- Résultat d'une opération (« expression ») avec XPath
soit un ou plusieurs nœuds sélectionnés dans la sous-arborescence
afin de les transformer dans un autre outil ²
soit la détermination d'une valeur à partir de sélection de nœuds
afin d'utiliser cette valeur dans un autre outil
- Exemples
tous les nœuds correspondant aux titres d'une référence de site
le texte du titre de la première référence de site
le nombre de prénoms d'auteur

XPATH EST EN FAIT UN COMPOSANT D'OUTILS ASSOCIÉS À XML
PRÉSENTATION ICI DE LA VERSION 1.0 DE XPATH

¹ En fait, XPath considère aussi chaque attribut comme un nœud mais avec un statut spécial.

² Cet outil autre intègre XPath en réutilisant le résultat obtenu ; exemples : XSLT, XQuery.

EXPRESSION DE XPATH : GÉNÉRALITÉS

FORMULATION PUISSANTE MAIS ASSEZ COMPLEXE

- PRINCIPALES CARACTÉRISTIQUES

- Expression à résultat de type variable
 - soit un ensemble de nœuds renvoyés par une sélection
 - soit une valeur (texte, nombre ¹ ou logique) issue d'un traitement appliqué à des nœuds à l'aide d'une formule de calcul
- Sélection notée sous la forme d'un « chemin » (*path*)
 - formulation avec plusieurs niveaux de décomposition
 - possibilité dans certains cas d'une écriture abrégée

- CHEMIN DE SÉLECTION

- Sélection progressive en plusieurs étapes de filtrage
 - les éléments issus d'une étape servent de base à l'étape suivante, exemple : les éléments de référence de livre puis leurs auteurs
 - chemin composé par la suite des notations de chaque étape en partant a priori de la racine du document
- Filtrage au niveau d'une étape
 - application d'un filtre à chaque élément de l'étape précédente qui en constitue la base de référence (son « contexte »)
 - exemple : les prénoms des auteurs précédemment sélectionnés
- Notation d'une étape constituée de deux niveaux obligatoires
 - a) une partie d'une arborescence définie par un « axe », exemple : tous les fils
 - b) un « critère » sélectionnant des nœuds selon leur nature, exemple : tous les éléments de titresoit la « localisation », avec un niveau complémentaire optionnel :
 - c) une ou plusieurs conditions de restriction appelées « prédicats », exemple : dont l'attribut `type-ref` est égal à `site`



PRÉSENTATION SIMPLIFIÉE DANS LE CADRE DE CETTE INTRODUCTION À XML

¹ Dans XPath, tous les nombres sont représentés sous la forme de réels (double précision de IEEE 754)

EXPRESSION DE XPATH : EXEMPLES INTRODUCTIFS

`/bibliographie /reference`

étape 1 : à partir de la racine du document, les éléments de bibliographie (1 seul)

étape 2 : puis tous ses éléments de référence

→ tous les éléments de référence dans la bibliographie

`/bibliographie /reference /child::*` ou `/bibliographie /reference /*`

étape 1 : à partir de la racine du document, les éléments de bibliographie (1 seul)

étape 2 : puis tous ses éléments de référence

étape 3 : puis tous les éléments fils de chaque élément de référence

→ tous les éléments fils d'un élément de référence

`//titre`

étape 1 : dans la descendance de la racine du document, les éléments de titre

→ tous les éléments de titre

`//reference /@id-ref`

étape 1 : parmi la descendance de la racine du document, les éléments de référence

étape 2 : puis l'attribut d'identification de chaque élément de référence

→ tous les identifications de références

`//reference[@type-ref = 'site'] /titre`

étape 1 : parmi la descendance de la racine du document, les éléments de référence
pour lesquels l'attribut de type correspond à un site

étape 2 : puis l'élément de titre de chacune de ces éléments de références

→ tous les titres de références de site

`//reference /auteur[2] /nom`

étape 1 : parmi la descendance de la racine du document, les éléments de référence

étape 2 : puis les deuxièmes éléments d'auteur d'une référence (à plus d'un auteur)

étape 3 : puis le nom de chacun de ces éléments d'auteur

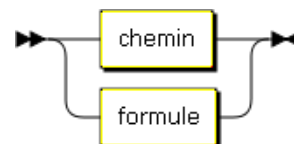
→ tous les noms de deuxième auteur de référence

EXPRESSION DE XPATH : FORME GÉNÉRALE

REPRÉSENTATION D'UNE SÉLECTION OU UN CALCUL

● NOTATION D'UNE EXPRESSION DANS XPATH

- Expression de calcul sous la forme d'une formule
résultat : valeur issue de l'évaluation de la formule
- Expression d'une sélection sous la forme d'un chemin
résultat : ensemble des nœuds sélectionnés par le chemin



● NOTATION D'UN CHEMIN

- Chemin absolu
sélection à partir du nœud ¹ représentant le document entier :
formule débutant par le préfixe `/`
sélection sur l'ensemble des éléments du document entier :
formule débutant par le préfixe `//` (descendance de la racine)
expression composée du préfixe suivi de la liste des étapes
- Chemin relatif
sélection à partir du nœud courant : formule ne débutant pas par `/`
expression composée de la liste des étapes
- Liste des étapes
soit une étape unique
soit plusieurs étapes avec `.` comme séparateur entre chaque étape :
 - soit le symbole `/` : l'étape suivante s'applique à chaque élément (qui devient le nœud courant) issu de l'étape précédente
 - soit le symbole `//` : l'étape suivante s'applique à la descendance de chaque élément issu de l'étape précédente
- Combinaison
union des nœuds de deux chemins : *chemin₁ | chemin₂*
- Espacements
possibilité de placer des blancs (espace, tabulation, retour à la ligne)
entre tous les éléments d'une formule, sauf à l'intérieur de `//`

RÉSULTATS D'UNE EXPRESSION, DE TYPE VARIABLE

¹ Cela correspond en fait au père de l'élément principal en XML

EXPRESSION DE XPATH : AXE ET CRITÈRE

LOCALISATION OBLIGATOIRE DANS UNE ÉTAPE : AXE + CRITÈRE

● AXE

- Indication d'une partie de l'arborescence du document à partir d'un nœud initial et selon une direction particulière

`ancestor::` les ancêtres
`ancestor-or-self::` les ancêtres plus le nœud considéré
`parent::` le parent (inexistant pour la racine du document)
`self::` le nœud considéré
`attribute::` ou `@` les attributs ¹ du nœud considéré
`child::` les fils
`descendant::` les descendants (fils et leurs descendants)
`descendant-or-self::` . les descendants plus le nœud considéré
`following::` les nœuds après dans le document ² sauf les descendants
`following-sibling::` ... les frères après dans le document ²
`preceding::` les nœuds avant dans le document ² sauf les ancêtres
`preceding-sibling::` ... les frères avant dans l'ordre fixé par le document ²

- Axe par défaut d'indication
ensemble des nœuds fils, c'est-à-dire `child::`

● CRITÈRE

- Désignation générique parmi l'arborescence de l'axe défini auparavant correspond à un sous-ensemble général de nœuds

nom d'élément les nœuds correspondant à cet élément
nom d'attribut un attribut particulier (axe des attributs)
*** tous les nœuds ou tous les attributs (selon l'axe)
`node()` tous les nœuds (éléments mais pas attributs)
`text()` les nœuds de texte
`comment()` les nœuds de commentaire
`processing-instruction()` .. les nœuds d'instructions de traitement ³

- Abréviations (axe + critère)

`//` `/descendant-or-self::node()` /
`.` `self::node()`
`..` `parent::node()`

¹ Les attributs sont des nœuds spéciaux dans Xpath, non considérés comme des fils du nœud de l'élément auquel ils sont rattachés.

² Ordre d'apparence des éléments dans le texte du document.

³ Une instruction de traitement (« PI ») se note par `<?nom ?>` mais ne correspond pas à `<?xml ?>` ; il est possible d'indiquer en paramètre de la notation le nom d'une instruction de traitement particulière, exemple : `processing-instruction('xml-stylesheet')`

EXPRESSION DE XPATH : FORMULE ET PRÉDICAT

● FORMULE

▪ Caractéristiques générales

une formule renvoie une valeur (texte, nombre réel, logique)
possibilité d'introduire des sous-formules (par parenthésage),
des valeurs, des variables ¹ et de faire appel à des fonctions
une formule peut débuter par un chemin (par exemple pour évaluer
l'existence de certains nœuds à partir du nœud en contexte)

▪ Manipulations à résultat logique (« booléen »)

comparaisons : `<` `<=` `>` `>=` `=` (égalité) `!=` (différence)
(sur des nombres, des textes ou des ensembles de nœuds)

combinaisons : `or` (ou) `and` (et) `not(expression)` (négation)

valeurs : `true()` (vrai) `false()` (faux)

et possibilité de conversion explicite : `boolean(expression)`

conversion implicite en une valeur fausse : ensemble de nœuds vide,
texte de longueur nulle, nombre nul

▪ Manipulation de textes

texte contenu ² dans un élément : *nom de l'élément*

valeur d'un attribut : *@nom de l'attribut*

chaîne de caractères notée entre guillemets (") ou apostrophes (')

▪ Calculs numériques

opérations : `+` `-` `*` `div` (division) `mod` (modulo)

notation d'un nombre (sans signe) : entier ou décimal

● PRÉDICAT

▪ Filtrage d'un ensemble de nœuds définis par un axe et un critère évaluation itérative pour chacun des nœuds de ce contexte

▪ Notation : `[formule]` à résultat logique ou entier

si logique : filtrage des nœuds pour lesquels la formule est vraie
(un texte en résultat de formule est converti en valeur logique :

`"Jules"` \rightarrow vrai, `" "` \rightarrow faux)

si entier non nul : extraction du nœud dont le rang est le résultat
(un nombre décimal en résultat de formule est tronqué : `1.7` \rightarrow 1)

possibilité d'indiquer plusieurs prédicats à la suite dans un filtrage

¹ Une variable se note par un \$ suivi d'un nom et se définit en fait dans un autre outil (XSLT, XQuery).

² C'est en fait le contenu du nœud de texte qui est le fils de l'élément considéré.

EXPRESSION DE XPATH : FONCTIONS

PRÉSENTATION DES PRINCIPALES FONCTIONS

● MANIPULATION D'UN NŒUD (OU D'UN ENSEMBLE)

`count (sélection)` : nombre de nœuds dans la sélection

`name (sélection)` : nom de l'élément ou de l'attribut résultat de la sélection ¹

`local-name (sélection)` : nom sans préfixe de l'élément ou de l'attribut de la sélection ¹

`position()` : position du nœud courant dans le contexte d'évaluation ²

`last()` : rang du dernier élément dans le contexte d'évaluation ²

`id (texte)` : élément identifié ³ par texte

● MANIPULATION DE TEXTES

`string (objet)` : conversion de *objet* en texte selon son type,

- valeur logique : `true` ou `false`

- nombre : notation décimale avec au-moins un chiffre avant, et au-moins un chiffre après le point décimal

- élément ² : texte de son contenu ou concaténation du texte de sa descendance

`substring (texte, rang, longueur)` : morceau de *texte* débutant au *rang* indiqué (compté à partir de 1) et de *longueur* caractères (ou jusqu'à la fin si non indiquée)

`concat (texte1, texte2)` : texte résultant de la concaténation de *texte₁* avec *texte₂*, possibilité d'ajouter d'autres paramètres (*texte₃*, etc.)

`contains (texte, motif)` : vrai si *texte* contient *motif* tel quel

`starts-with (texte, motif)` : vrai si *texte* débute par *motif* tel quel

`normalize-space (texte)` : renvoi de *texte* après élimination des blancs ⁴ initiaux et finaux, et remplacement de plusieurs blancs internes successifs par un seul espace

`string-length (texte)` : nombre de caractères dans *texte*

● MANIPULATION DE VALEURS NUMÉRIQUES

`number (objet)` : conversion de *objet* en nombre selon son type,

- valeur logique : 1 pour vrai, 0 pour faux

- texte : si le texte correspond à la notation d'un nombre entier ou décimal

(c'est-à-dire un signe éventuel suivi de chiffres, avec un éventuel point décimal, et avec d'éventuels espaces au début et/ou à la fin) alors valeur numérique en résultat, sinon une valeur indéfinie (NaN, *not a number*)

- nœud ² : converti en texte puis en nombre

`sum (sélection)` : somme des valeurs des éléments de la sélection

`round (nombre)` : valeur arrondie du nombre

¹ Si la sélection renvoie plusieurs nœuds, il s'agit alors du 1^{er} selon l'ordre fixé par le document ; en cas d'absence de paramètre (dans un prédicat), c'est alors le nœud courant du contexte qui est considéré.

² Utilisation dans un prédicat appliqué à un ou plusieurs nœuds (le contexte d'évaluation du prédicat).

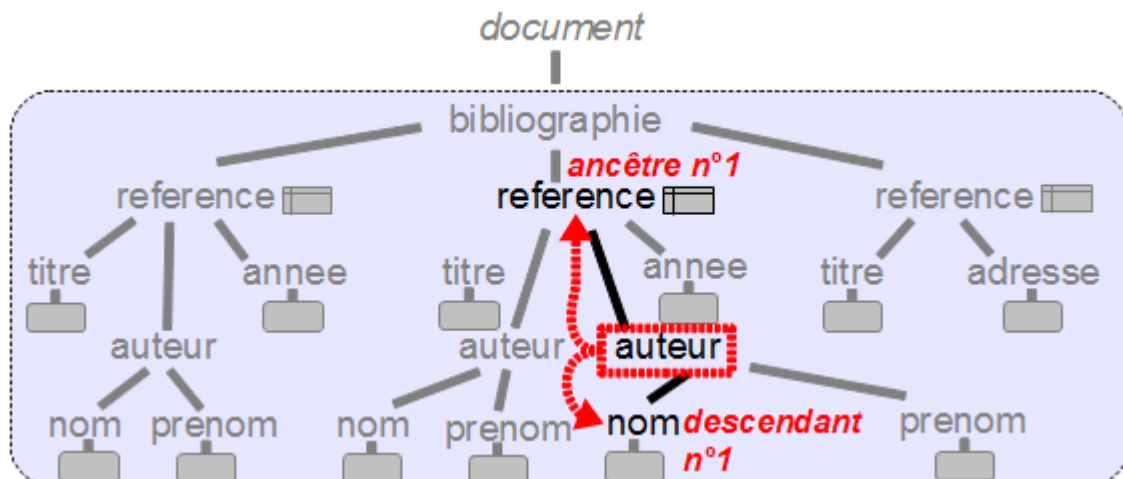
³ Identification via un attribut déclaré avec le type `ID` dans la DTD.

⁴ Un blanc correspond à un espace, une tabulation ou un retour à la ligne.

EXPRESSION DE XPATH : NŒUDS ET ÉVALUATION

- ENSEMBLE DE NŒUDS

- Nature des nœuds
en général un nœud d'élément, plus rarement un nœud d'attribut
- Origine d'un ensemble de nœuds
soit le contexte d'un prédicat défini par une localisation (axe et critère)
soit le résultat d'une localisation placée au début d'une formule
- Rang d'un nœud
comptage à partir de 1 selon l'ordre de la direction de l'axe considéré
relativement au nœud considéré
soit a priori selon l'ordre d'apparition dans le document
soit l'inverse pour l'axe des ancêtres ou des frères aînés



- APPLICATION D'UN PRÉDICAT

- Evaluation itérative
contexte d'évaluation formé des nœuds de la localisation courante
prédicat évalué successivement pour chaque nœud du contexte
- Exemple

```
//* [ text() [ contains(., "XML") ] ]
```


pour chacun des nœuds de l'arborescence ... `//*`
tester s'il existe un nœud de texte fils `[text()]`
qui contient le motif "XML" `[contains(., "XML")]`

EXPRESSION DE XPATH : CONVERSIONS

CARACTÉRISTIQUES GÉNÉRALES

- CONVERSIONS POSSIBLES

conversion ↙	élément	attribut	texte	logique	nombre
texte	texte de son contenu ou concaténation du texte de sa descendance	valeur		true ou false	notation décimale avec au-moins un chiffre avant, et au-moins un chiffre après le point décimal
logique	vrai si existant, faux sinon	vrai si existant, faux sinon	vrai si texte de longueur non nulle, faux sinon		faux si nul, vrai sinon
nombre	conversion en texte, puis de ce texte en nombre	conversion du texte de sa valeur	valeur numérique si le texte correspond à la notation d'un nombre, sinon valeur indéfinie (NaN)	1 pour vrai, 0 pour faux	

- MODALITÉS D'APPLICATION

- Conversion explicite

`string(objet)` : conversion de *objet* en texte

`boolean(objet)` : conversion de *objet* en valeur logique

`number(objet)` : conversion de *objet* en nombre

- Conversion implicite

prédicat : conversion d'un résultat de sa formule ni entier ni logique, soit en nombre entier si nombre décimal, soit sinon en valeur logique

formule : conversion selon le type attendu par l'opération (par exemple texte en valeur logique pour effectuer une combinaison logique « ou »)

fonction : conversion selon le type attendu en paramètre, avec prise en compte du 1^{er} nœud d'un ensemble si une seul nœud attendu

EXPRESSION DE XPATH : EXEMPLES AVANCÉS

`/*`

élément principal (fils unique de la racine du document)

`//annee /preceding-sibling::auteur`

les éléments des auteurs d'une référence avec une année

`//reference[@type-ref = "livre"] /auteur[2] /nom`

les éléments du nom des deuxièmes auteurs de livre

`//reference[count(auteur) > 1]`

les éléments des références avec plus d'un auteur

`//reference[annee = "2000"]`

les éléments des références pour l'année 2000

`//auteur[contains(prenom, "Paul")]`

les éléments des auteurs dont le prénom contient Paul

`//reference[(annee >= 2005) and (annee <= 2009)] /titre`

les éléments de titre des références d'année comprise entre 2005 et 2009

`//reference[last()]`

l'élément de la dernière référence

`//reference[not (auteur)]`

tous les nœuds des références sans auteur

`//reference /@*`

tous les attributs des nœuds de références

`count(//reference)`

le nombre de références

`. //text()`

les nœuds de texte descendants du nœud courant

EXPRESSION DE XPATH : ESPACE DE NOMMAGE

PRISE EN CHARGE DES ESPACES DE NOMMAGE DANS XPATH 1.0

- RECONNAISSANCE PARTIELLE

- Identification de l'espace de nommage par son URI
mais sans reconnaissance automatique du préfixe (attribut `xmlns:`),
impossibilité a priori d'utiliser un nom qualifié (ex. : `bib:bibliotheque`)
- Déclaration manuelle du préfixe d'un espace de nommage
mécanisme spécifique à l'environnement d'évaluation utilisé,
nécessaire pour le nommage complet d'un élément dans une expression
exemple d'expression valide après déclaration du préfixe `bib:` d'espace
de nommage (désigné par son URI) : `//bib:bibliotheque`

- NOTATIONS SPÉCIFIQUES

- Axe des espaces de nommage

notation : `namespace::`

exemple : `count(//namespace::*[. = 'mailto:auteur@ici.fr'])`

le nombre d'éléments dans l'espace de nommage 'mailto:auteur@ici.fr'

- Fonctions

`local-name (sélection)` : nom du nœud ¹ sans préfixe

exemple : `//* [local-name(.) = 'bibliotheque']`

A noter : la fonction `name()` devrait renvoyer le libellé complet d'un nom qualifié
mais cela n'est de fait pas garanti dans tous les environnements d'exécution

exemple : `name(//bib:bibliotheque[1])` peut s'évaluer selon les cas en
'bib:bibliotheque' ou 'bibliotheque'

`namespace-uri (sélection)` : URI de l'espace de nommage d'élément ou attribut ¹

exemple : `namespace-uri(//bib:bibliotheque[1])`

l'URI de l'espace de nommage du 1^{er} élément `bibliotheque` dans le document
(`'mailto:auteur@ici.fr'`)

MODIFICATIONS APPORTÉES PAR LA VERSION 2 DE XPATH

¹ Si la sélection renvoie plus d'un nœud, il s'agit alors du 1^{er} selon l'ordre fixé par le document

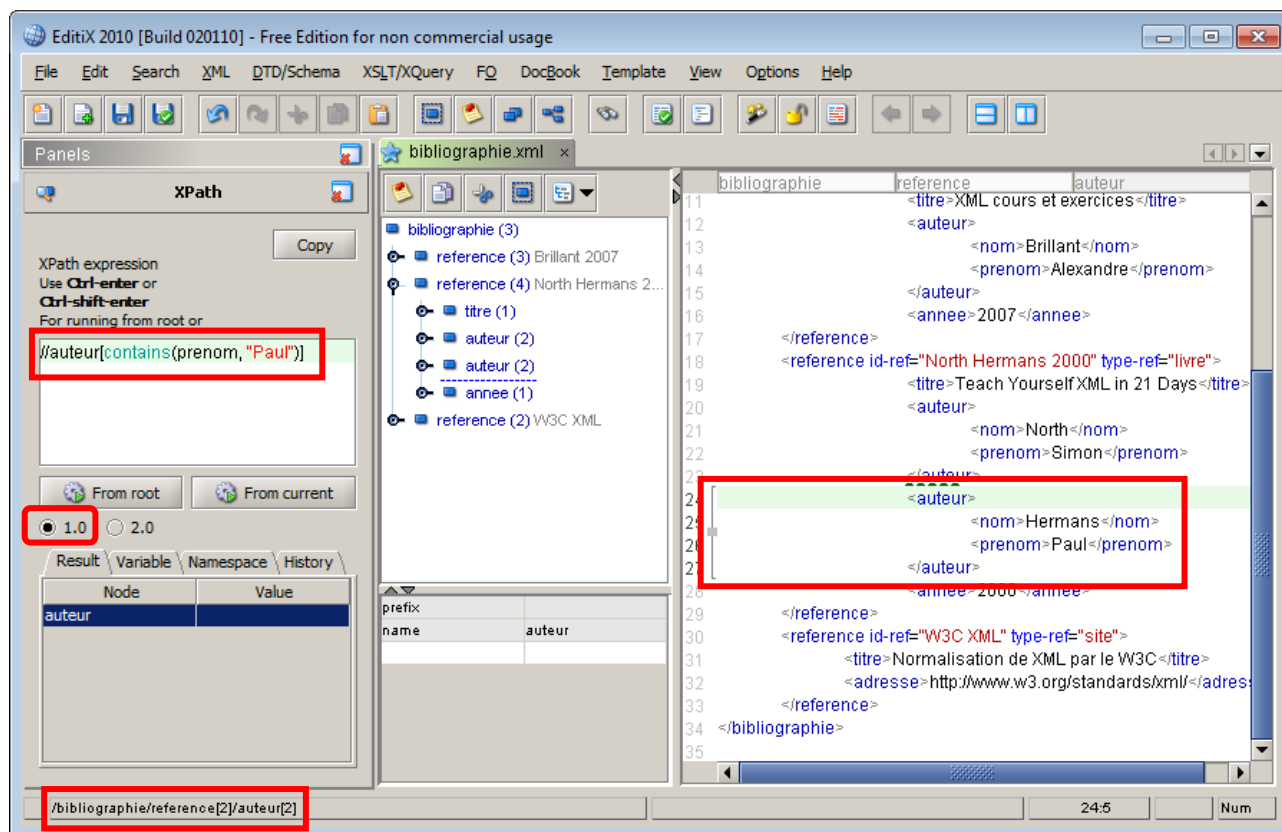
MANIPULATION DE XPATH AVEC EDITIX

CAS DE LA VERSION 2010 D'EDITIX POUR WINDOWS

▪ Panneau dédié à XPath

commande VIEW WINDOWS XPATH VIEW

vérification de la version de XPath utilisée (ici 1.0)



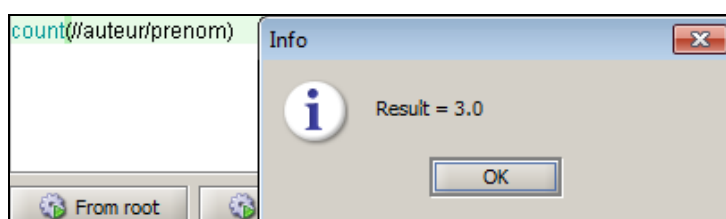
a) Saisie de l'expression dans le panneau
exemples :

- 1) `//auteur[contains(prenom, "Paul")]`
- 2) `count(//auteur/prenom)`

b) Activation soit à partir de la racine, soit à partir du nœud courant

c) Eléments sélectionnés signalés dans les panneaux et en bas,
ou affichage de la valeur calculée
exemples :

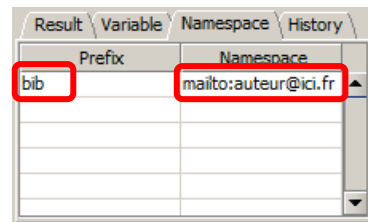
- 1) `/bibliographie/reference[2]/auteur[2]`
- 2)



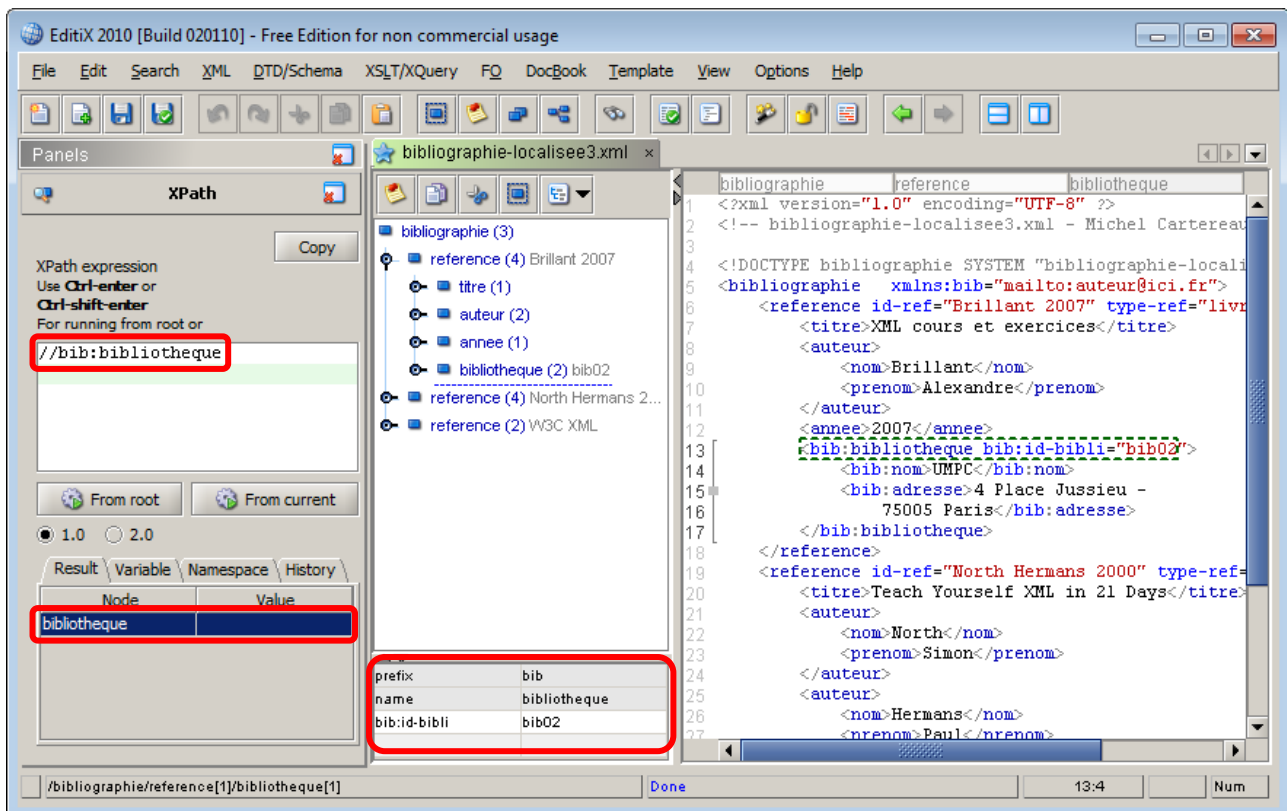
XPATH ET ESPACES DE NOMMAGES AVEC EDITIX

CAS DE XPATH 1.0

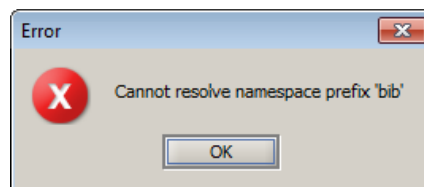
- Déclaration d'un espace de nommage
 - activation de l'onglet NAMESPACE dans le panneau de XPATH
 - saisies du préfixe et de l'URI pour chaque espace de nommage



- Exemple d'expression avec nommage qualifié d'élément
 - déclaration obligatoire de l'espace de nommage au préalable



Message d'erreur en cas d'espace de nommage non déclaré



EXERCICES SUR LES EXPRESSIONS DE XPATH

1. Dans l'exemple de la bibliographie, écrire les expressions pour :

- a) les éléments d'auteur
- b) le nombre de références de site
- c) les noms des auteurs de références de l'année 2007
- d) le nom du premier auteur
- e) le premier auteur de la première référence de livre
- f) les ancêtres de l'auteur au prénom contenant Paul
- g) les frères de la deuxième référence
- h) les identifications des références de livre

2. Est-ce que les expressions suivantes sont valides :

- a) `15 * 99`
- b) `//*[2]`
- c) `count(//prenom) >= 1`
- d) `//titre /ancestor::reference[@type-ref = "site"]`
- e) `sum(//reference[@type-ref = "livre"] / annee)`
- f) `//reference[3 = count(//auteur)] / auteur / nom`

3. Est-il possible de déterminer :

- a) la taille totale des textes de titre de livre ?
- b) s'il existe une référence de site située juste après une de livre ?

TRANSFORMATION D'UN DOCUMENT AVEC XSLT

GÉNÉRALITÉS

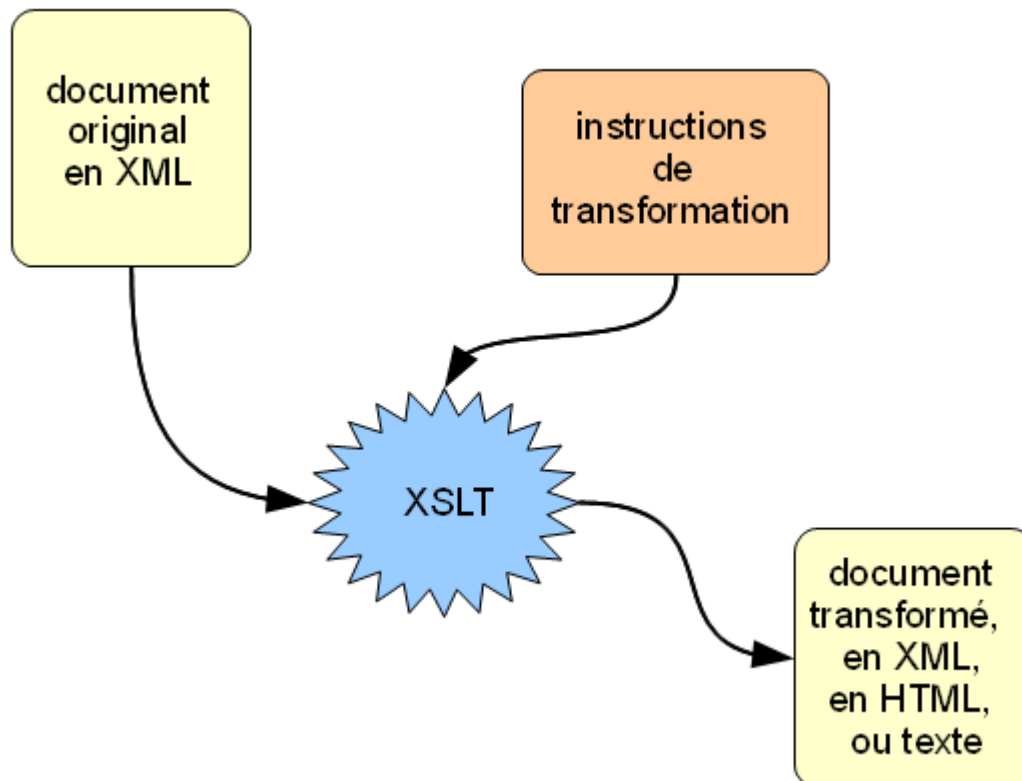
MODÈLE DE TRANSFORMATION

PRINCIPALES INSTRUCTIONS

EXEMPLES DE FEUILLES DE STYLE

PRÉSENTATION DE XSLT

OUTILS DE TRANSFORMATION DE TEXTE (*EXTENSIBLE STYLESHEET LANGUAGE TRANSFORMATIONS*)



- PRINCIPALES CARACTÉRISTIQUES

- Transformations du texte d'un document en XML par un « processeur » production d'une version en XML, HTML ou autre format de texte
autre outil XSL-FO ¹ pour des formats plus évolués (PDF, PS, ...)
- Description des transformations
document annexe (feuille de style, *stylesheet*) composé par des instructions de traitement notées dans le langage de XSLT ²
transformations basées sur des modèles (*template*) associés à des nœuds du document original décrit par le formalisme XPath

PRÉSENTATION ICI MINIMALE DE LA VERSION 1.0 DE XSLT

¹ XSL-FO : *extensible stylesheet language - formatting objects*

² Langage hérité de *Document style semantics and specification language* (DSSSL) de SGML, qui était basé sur le langage de programmation fonctionnelle Scheme ; cela correspond à un mode de programmation déclaratif et non pas impératif (comme C, Java etc.).

XSLT : EXEMPLE DE LA BIBLIOGRAPHIE

PAGE WEB PRODUITE DE MANIÈRE AUTOMATIQUE



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="fr" lang="fr">
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1"></meta>
  <title>bibliographie</title>
</head>
<body>
  <h1>bibliographie</h1>
  <dl>
    <dt>[Brillant 2007]</dt>
    <dd>Brillant Alexandre.
      <strong>XML cours et exercices</strong>. 2007.</dd>
    <dt>[North Hermans 2000]</dt>
    <dd>North Simon, Hermans Paul.
      <strong>Teach Yourself XML in 21 Days</strong>. 2000.</dd>
    <dt>[W3C XML]</dt>
    <dd><strong>Normalisation de XML par le W3C</strong>.
      En ligne &agrave; :
      <a href="http://www.w3.org/standards/xml/">
        http://www.w3.org/standards/xml/</a>.
    </dd>
  </dl>
</body>
</html>
```

LANGAGE XSLT : SQUELETTE

UN LANGAGE ÉCRIT EN XML

- SQUELETTE D'UNE FEUILLE DE STYLE

- Utilisation d'un espace de nommage spécifique
préfixage des noms de balise par `xsl:`

- Structuration générique

élément principal de la feuille de style : `<xsl:stylesheet>` ¹ englobant
l'élément indiquant la nature du document produit : `<xsl:output />`
puis un ou plusieurs modèles de transformations : `<xsl:template>`

- Exemple :

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="html" encoding="UTF-8"
    doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
    doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>

  <xsl:template ... >
    [...]
  </xsl:template>

  [...]

</xsl:stylesheet>
```

- BALISE `<xsl:output/>`

- Caractéristiques générales du document produit
en HTML, XML ou en texte brut sans aucune balise (y compris d'HTML)

- Principaux attributs

<code>method</code>	type de document produit (xml ou html ou text)
<code>encoding</code>	encodage du texte produit (UTF-8 en général)
<code>doctype-public</code>	valeur pour la balise DOCTYPE (cas du type html)
<code>doctype-system</code>	valeur pour la balise DOCTYPE (cas du type html)
<code>media-type</code>	type MIME ² du document (par défaut : text/html pour html, text/xml pour xml, text/plain pour text)
<code>indent</code>	indentation des balises (yes) ou non (no), si possible (par défaut : yes pour html, no pour xml)

¹ Élément principal synonyme de `<xsl:stylesheet>` : `<xsl:transform>`

² MIME : typologie de contenu notée « *type/sous-type* », et définie par le W3C (RFC 2045 à 2049)

LANGAGE XSLT : MODÈLE DE TRANSFORMATION

COMPOSANT ESSENTIEL DE LA FEUILLE DE STYLE

- DÉFINITION D'UN MODÈLE DE TRANSFORMATION

- Ensemble d'instructions de traitements

transformation d'un nœud désigné par une expression de XPath

et aussi éventuellement d'une partie de l'arborescence du document

exemple : un nœud de référence et toute sa descendance

- Balise de définition `<xsl:template match="motif">`

attribut `match` avec pour valeur un motif (*pattern*) exprimé en XPath,

indiquant à quel type de nœuds s'applique ce modèle de transformation,

et noté sous la forme d'un chemin de localisation de XPath où les seuls

axes possibles sont les fils (`child:.`) et les attributs (`attribute:.`)

modèle appliqué à un nœud N si et seulement s'il existe un contexte où

l'évaluation de la sélection renvoie un ensemble de nœuds contenant N

(contexte alors constitué du nœud N ou d'un de ses ancêtres)

- Contenu

texte à recopier en sortie entremêlé avec des instructions de

transformation du langage XSLT insérées sous forme de balises

exemple : réécriture en HTML du texte complet de la référence composé

de son titre, de son ou ses auteurs, et de son année de publication

- Exemple

modèle de transformation pour un élément d'auteur

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

```
  <xsl:value-of select="prenom"/> <xsl:value-of select="nom"/>
```

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

production d'une ligne avec le prénom suivi du nom (`<xsl:value-of>`)

```
<auteur>
```

```
  <nom>Brillant</nom>
```

```
  <prenom>Alexandre</prenom>
```

```
</auteur>
```



Alexandre Brillant

LANGAGE XSLT : MODÈLE DE TRANSFORMATION (BIS)

APPLICATION AUTOMATIQUE DES MODÈLES

- APPLICATION D'UN MODÈLE DE TRANSFORMATION

- Activation du processus de transformation

- démarrage par application automatique à la racine du document,
puis application récursive au sein d'un modèle à des éléments choisis

- Balise d'application `<xsl:apply-templates select="sélection">`

- attribut `select` sous la forme d'une expression de XPath, qui correspond à un sous ensemble de nœuds sélectionnés à partir du nœud courant, à transformer récursivement à l'aide d'un modèle

- en cas d'absence de l'attribut `select`, application automatique à tous les nœuds fils du nœud courant, y compris les nœuds de texte

- tri possible du sous-ensemble de nœuds (cf. page 58)

- Exemple

- modèle de transformation pour la liste des auteurs

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="text" encoding="UTF-8" />

  <!-- depart de la transformation (racine) -->
  <xsl:template match="/">
    <xsl:text>Auteurs de la bibliographie<xsl:text>"
    <xsl:apply-templates select="//auteur"/>
  </xsl:template>

  <!-- modele de transformation pour un auteur -->
  <xsl:template match="auteur">
    <xsl:value-of select="prenom"/> <xsl:value-of select="nom"/>
  </xsl:template>

</xsl:stylesheet>
```

AU-MOINS UN MODÈLE À DÉFINIR (A PRIORI POUR LA RACINE DU DOCUMENT)

LANGAGE XSLT : ALGORITHME DE TRANSFORMATION

PRÉSENTATION SIMPLIFIÉE

- CARACTÉRISTIQUES GÉNÉRALES

- Algorithme principal

```
début
  transformer(racine du document)
fin

procédure transformer(noeud)
  si (au-moins un modèle défini1 s'applique à noeud) alors
    si plusieurs modèles concurrents alors
      choisir un des modèles selon les règles de priorité2
    fin du si
    application du modèle défini au noeud
  sinon
    application d'un modèle implicite
  fin du si
fin de la procédure
```

- Transformation potentiellement récursive

possibilité lors de l'application d'un modèle défini ou implicite, d'un appel récursif de la procédure `transformer()` pour d'autres nœuds

- Modèle implicite

modèle appliqué par défaut d'existence d'un modèle défini approprié
application récursive à la racine et à tous les éléments du document

```
<xsl:template match="*" | "/" >
  <xsl:apply-templates />
  <!-- application aux noeuds fils, y compris de texte -->
</xsl:template>
```

recopie des nœuds de texte et de toutes les valeurs d'attributs

```
<xsl:template match="text() | @" >
  <xsl:value-of select="." />
</xsl:template>
```

pas de recopie des commentaires et des instructions de traitement

```
<xsl:template match="comment() | processing-instruction()" />
```

¹ C'est-à-dire un modèle défini explicitement dans la feuille de style avec l'instruction `<xsl:template>`

² Règles de priorité non présentées ici dans cette introduction.

LANGAGE XSLT : PRODUCTION DE TEXTE

PRÉSENTATION DES INSTRUCTIONS ESSENTIELLES

● PRODUCTION DE TEXTE

- Reproduction de tout texte étranger à XSLT présent au sein d'un modèle
tout ce qui n'est pas une balise de XSLT est reproduit tel quel (ex. : `<h1>`)
- Texte issu du document : `<xsl:value-of select="sélection" />`
sélection via une expression de XPath, de résultat converti en texte
exemple : `<xsl:value-of select="name(/*)" />`
- Cas du texte d'une sélection transformé en valeur d'attribut ¹ : `{sélection}`
forme abrégée de `<xsl:value-of />`
exemple : ``
- Texte littéral : `<xsl:text> texte </xsl:text>`
utile pour garantir le respect des espaces du texte
exemple : `<xsl:text> </xsl:text>`
- Commentaire produit (`<!-- -->`) : `<xsl:comment> texte 2 </xsl:comment>`
exemple : `<xsl:comment> Transformation via XSLT </xsl:comment>`

● GESTION DES BLANCS ³

- Cas des blancs dans la feuille de style
seulement conservés au sein de `<xsl:text>`
- Cas des blancs dans les noeuds de texte du document originel
blancs au sein de texte (entre les mots) reproduits a priori ⁴ tel que
cas particulier : texte avec uniquement des blancs (noeud de texte blanc)
- Cas des noeuds de texte blancs dans le document originel
a priori conservés et reproduits tel que
possibilité d'élimination via `<xsl:strip-space elements="liste" />`
où liste contient soit un ou plusieurs noms d'éléments séparés par
un blanc, soit le symbole `*` pour désigner tous les éléments
et en complément, annulation de l'élimination et donc conservation, via
`<xsl:preserve-space elements="liste" />`
exemple :
`<xsl:strip-space elements="*" />`
`<xsl:preserve-space elements="remarques" />`

¹ Il s'agit du cas où il n'est pas possible de placer la balise `<xsl:value-of>` car on se trouve déjà dans la définition d'une autre balise, au niveau du texte d'un attribut.

² Par précaution, entourer le texte d'au moins un blanc afin qu'il ne colle pas à la balise produite (`<!-- -->`).

³ Caractères blancs : espace (20_h), tabulation (09_h), saut-de-ligne (0A_h) et retour-à-la-ligne (0D_h).

⁴ Autres possibilités avec les fonctions de XPath : `normalize-space()`, `translate()` ou `replace()`

LANGAGE XSLT : VARIABLE, CHOIX

● VARIABLE

- Valeur réutilisable dans les balises de XSLT
valeur constante associée à un nom de variable et non modifiable,
calculée par une évaluation au moment de sa définition,
réutilisable dans une expression à l'aide de la notation : `&nom` mais
seulement au sein de l'élément de XSLT où la variable est définie ¹
- Cas de définition comme valeur issue d'une expression de XPath
notation : `<xsl:variable name="nom" select="expression" />`
avec utilisation d'apostrophes (« ' ») pour délimiter un texte,
valeur à texte vide en cas d'absence de l'attribut `select`
exemple : `<xsl:variable name="couleur" select="'green'" />`
- Cas de définition comme texte issue d'une transformation en XSLST
notation : `<xsl:variable name="nom"> code en XSLST </xsl:variable>`
exemple :
`<xsl:variable name="nom-normalise" >`
`<xsl:value-of select="normalize-space(name(/*))" />`
`</xsl:variable>`
et réutilisation par : `<xsl:value-of select="$nom-normalise" />`

● CHOIX

- Instruction à choix unique : `<xsl:if test="condition">`
contenu traité si la condition (expression de XPath) est vérifiée
(sans possibilité d'exprimer l'alternative, « sinon »)
exemple : `<xsl:if test="auteur"> Auteur </xsl:if>`
- Instruction à choix multiple : `<xsl:choose>`
contient une suite de choix correspondant chacun à une condition
un choix est noté par : `<xsl:when test="condition"> </xsl:when>` et
contient une suite d'instructions traitée si la condition est vérifiée
le premier choix à la condition vérifiée est exécuté et seulement celui-ci
si aucun choix n'est vérifié, une instruction d'alternative est alors
exécutée si présente : `<xsl:otherwise> </xsl:otherwise>`
exemple :
`<xsl:choose>`
`<xsl:when test=" @type-ref = 'site' ">Site</xsl:when>`
`<xsl:when test=" @type-ref = 'livre' ">Livre</xsl:when>`
`<xsl:otherwise>?</xsl:otherwise>`
`</xsl:choose>`

¹ Portée de la variable limitée à l'élément et ses sous-éléments en XSLT.

LANGAGE XSLT : ITÉRATION, TRI

● ITÉRATION

- Répétition « pour chaque » : `<xsl:for-each select="sélection">`
instructions répétées pour chaque élément de la sélection définie sous la forme d'une expression de XPath qui correspond à un sous ensemble de nœuds sélectionnés à partir du nœud courant
à l'intérieur de cette balise, le contexte courant est automatiquement fixé à l'élément courant de la sélection lors de chaque itération
tri possible du sous-ensemble de nœuds (cf. ci-après)

exemple :

```
<!-- liste des noms de chaque auteur -->
<xsl:for-each select="auteur">
  <xsl:value-of select="nom" />
</xsl:for-each>
```

● TRI

- Ordonnancement d'un ensemble d'éléments
ceux de la sélection dans une itération ou l'application d'un modèle
- Tri des éléments : `<xsl:sort select="clef" options />`
placement à l'intérieur de la balise d'itération ou d'application de modèle
avec possibilités de tris emboîtés par répétition de la balise
clef de tri définie sous la forme d'une expression de XPath,
appliquée à chaque élément du contexte, et convertie en texte

options possibles :

- `order = "sens"` : ordre croissant (`ascending`, par défaut d'indication) ou décroissant (`descending`)
- `data-type = "mode"` : tri lexicographique (`text`, par défaut d'indication) ou numérique (`number`, clef convertie en nombre)
- `case-order = "code"` : majuscule classée avant minuscule (`upper-first`) ou l'inverse (`lower-first`)

exemples :

```
<xsl:for-each select="auteur">
  <!-- tri des auteurs selon le nom, puis le prenom -->
  <xsl:sort select="nom" /> <xsl:sort select="prenom" />
  <xsl:value-of select="nom" />
</xsl:for-each>
```

ou au niveau de l'application d'un modèle :

```
<xsl:apply-templates select="//auteur"/>
  <xsl:sort select="nom" /> <xsl:sort select="prenom" />
</xsl:apply-templates>
```

LANGAGE XSLT : CRÉATION D'ÉLÉMENT, D'ATTRIBUT

● CRÉATION D'ÉLÉMENT

- **Définition d'un élément :** `<xsl:element name="nom d'élément" >`
production d'un nouvel élément avec le nom indiqué
attribution facultative d'un espace de nommage : `namespace="espace"`
- **Exemples**

```
<!-- identification transformee en element "code" -->
<xsl:element name="code">
  <xsl:value-of select="@id-ref" />
</xsl:element>
```

ou en utilisant Dublin core :

```
<!-- identification transformee en element DC -->
<xsl:element name="dc:identifier"
  namespace="dc:http://purl.org/dc/terms/">
  <xsl:value-of select="@id-ref" />
</xsl:element>
```

● CRÉATION D'ATTRIBUT

- **Définition d'un attribut :** `<xsl:attribute name="nom d'attribut" >`
attribut ajouté avec le nom indiqué (remplacement si déjà existant)
attribution facultative d'un espace de nommage : `namespace="espace"`
- **Déclaration d'un jeu d'attributs :** `<xsl:attribute-set name="nom de jeu" >`
ensemble de définition d'attributs identifié par un nom de jeu et à placer
au 1^{er} niveau de la feuille de style (fils de `<xsl:stylesheet>`)
appliqué à un élément via l'attribut : `use-attribute-sets="nom de jeu..."`
- **Exemples**

```
<!-- ajout de l'attribut "type-code" -->
<xsl:element name="code">
  <xsl:attribute name="type-code">perso</xsl:attribute>
  <xsl:value-of select="@id-ref" />
</xsl:element>
```

et pour un jeu d'attributs :

```
<!-- definition du jeu d'attributs "codage" au 1er niveau -->
<xsl:attribute-set name="codage">
  <xsl:attribute name="type-code">perso</xsl:attribute>
  <xsl:attribute name="version">1</xsl:attribute>
</xsl:attribute-set>
```

[...]

```
<!-- utilisation du jeu d'attributs -->
<xsl:element name="code" use-attribute-sets="codage" >
  <xsl:value-of select="@id-ref" />
</xsl:element>
```

FEUILLE DE STYLE EN XSLT DE L'EXEMPLE

FICHIER bibliographie-xhtml.xsl

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

<!-- Michel Cartereau 19 IX 2013 -->

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="html" encoding="UTF-8" indent="yes"
    doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
    doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>

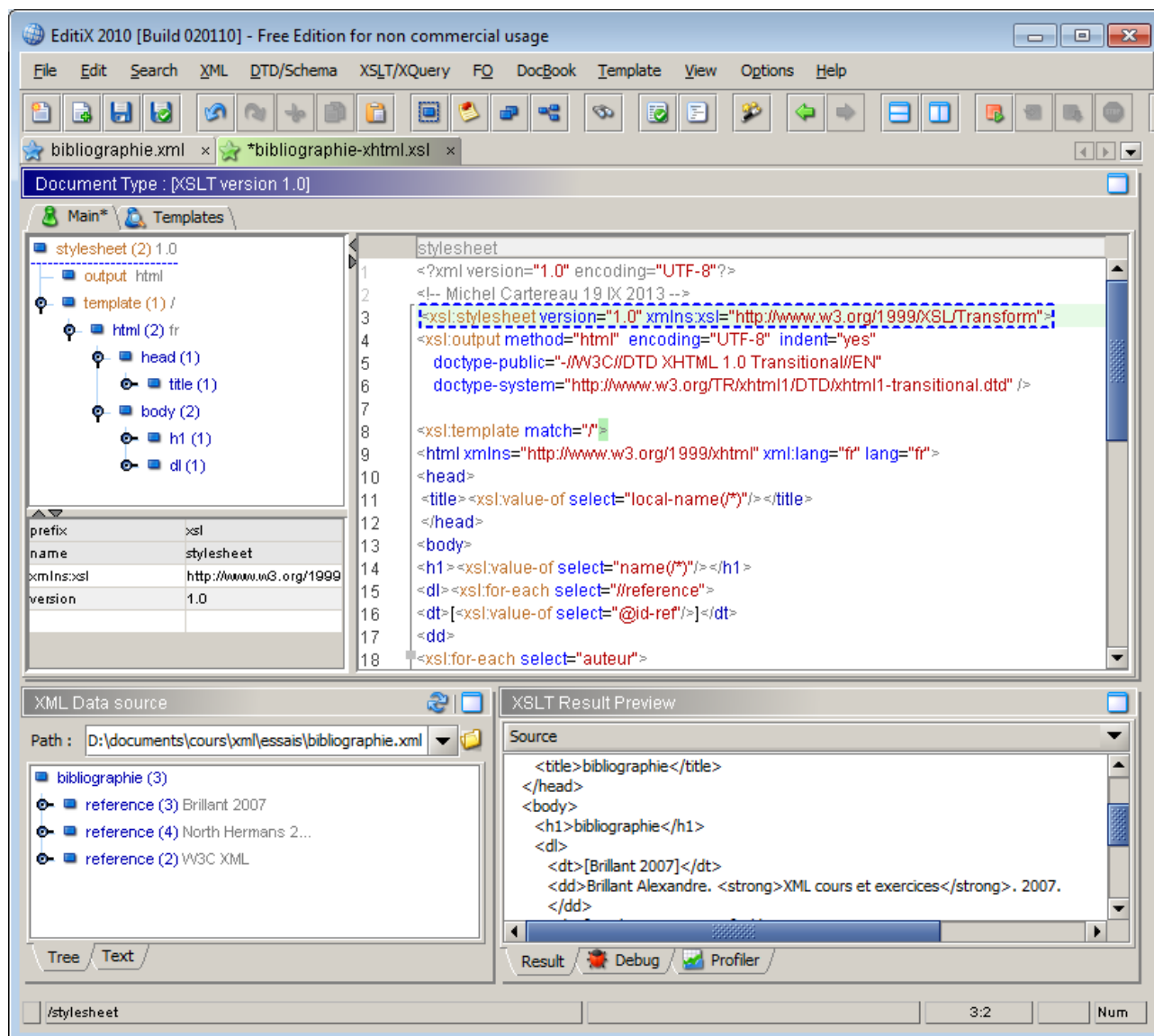
  <xsl:template match="/">
    <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
      <head>
        <title><xsl:value-of select="name(/*)" /></title>
      </head>
      <body>
        <h1><xsl:value-of select="name(/*)" /></h1>
        <dl><xsl:for-each select="//reference">
          <dt>[<xsl:value-of select="@id-ref"/>]</dt>
          <dd>
            <xsl:for-each select="auteur">
              <xsl:if test="position() > 1">, </xsl:if>
              <xsl:value-of select="nom"/>
              <xsl:if test="prenom">
                <xsl:text> </xsl:text><xsl:value-of select="prenom"/>
              </xsl:if>
            </xsl:for-each>
            <xsl:if test="count(auteur) > 0">. </xsl:if>
            <strong><xsl:value-of select="titre"/></strong><xsl:text>. </xsl:text>
            <xsl:if test="adresse">
              En ligne à : <a href='{adresse}'><xsl:value-of select="adresse"/></a>
              <xsl:text>. </xsl:text>
            </xsl:if>
            <xsl:if test="annee">
              <xsl:value-of select="annee"/> <xsl:text>. </xsl:text>
            </xsl:if>
          </dd>
        </xsl:for-each></dl>
      </body>
    </html>
  </xsl:template>

</xsl:stylesheet>
```

TRANSFORMATION EN XSLT AVEC EDITIX

EDITION DU FICHIER `bibliographie-xhtml.xml`

Commande XSLT/XQUERY, TRANSFORM A DOCUMENT WITH THIS XSLT... ¹



panneau avec vue sur le texte produit en résultat de la transformation

Autre possibilité : transformation automatique dans un navigateur en ajoutant au document XML, l'indication de la feuille de style en XSL :

```
<?xml-stylesheet type="text/xsl" href="bibliographie-xhtml.xml" ?>
```

¹ Si la commande de transformation n'est pas activable, c'est sans doute un mauvais type attribué au document par Editix ; fixer alors le type du fichier à XSLT via la commande FILE FILE INFO

EXEMPLE DE TRANSFORMATION EN XML

PRODUCTION DE LA LISTE DES IDENTIFIANTS DE RÉFÉRENCE EN XML

● EXEMPLE DE RÉSULTAT

```
<?xml version="1.0" encoding="UTF-8"?>
<liste-id source="bibliographie">
  <id>Brillant 2007</id>
  <id>North Hermans 2000</id>
  <id>W3C XML</id>
</liste-id>
```

● FEUILLE DE STYLE EN XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- bibliographie-id-xml.xsl - Michel Cartereau 20 IX 2013 -->
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="xml" indent="yes" encoding="UTF-8" />

  <!-- depart de la transformation -->
  <xsl:template match="/">
    <liste-id source="{name(/*)}">
      <!-- transformation pour chaque attribut id-ref -->
      <xsl:apply-templates select="//@id-ref"/>
    </liste-id>
  </xsl:template>

  <!-- transformation pour un attribut id-ref -->
  <xsl:template match="@id-ref">
    <id><xsl:value-of select="."/></id>
  </xsl:template>

</xsl:stylesheet>
```

● ASTUCE

■ Mise en forme automatique

utilisation de l'attribut `indent` de `<xsl:output>` pour une indentation réalisée automatiquement lors de la transformation

exemple :

```
<xsl:output method="xml" indent="yes" encoding="UTF-8" />
```

EXEMPLE DE TRANSFORMATION EN TEXTE BRUT

PRODUCTION DE LA LISTE DES AUTEURS EN TEXTE BRUT

● EXEMPLE DE RÉSULTAT

```
Auteurs dans "bibliographie"
- Brillant Alexandre
- North Simon
- Hermans Paul
```

● FEUILLE DE STYLE EN XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- bibliographie-auteurs-txt.xsl - Michel Cartereau 20 IX 2013 -->
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="text" encoding="UTF-8" />

  <!-- depart de la transformation -->
  <xsl:template match="/">
    <xsl:text>Auteurs dans "</xsl:text>
    <xsl:value-of select="name(/*)" />
    <xsl:text>"
  </xsl:text>
  <xsl:apply-templates select="//auteur"/>
  </xsl:template>

  <!-- transformation pour un element auteur -->
  <xsl:template match="auteur">
    <xsl:text>- </xsl:text><xsl:value-of select="nom"/>
    <xsl:if test="prenom">
      <xsl:text> </xsl:text><xsl:value-of select="prenom"/>
    </xsl:if>
    <xsl:text>
  </xsl:text>
  </xsl:template>

</xsl:stylesheet>
```

● ASTUCE

- Contrôle des sauts de ligne dans le texte produit
 - utilisation systématique de la balise `<xsl:text>` pour produire du texte
 - insertion d'un saut de ligne comme un texte réduit à ce saut de ligne
 - ou bien au(x) caractère(s) de contrôle équivalent(s) ¹
- exemples : `<xsl:text>
</xsl:text>`
`<xsl:text>`
`</xsl:text>`

¹ Saut à la ligne dans Windows par ASCII n° 13 (CR) puis n° 10 (LF), dans Linux par ASCII n° 10 (LF)

EXERCICES SUR XSLT

1. Dans l'exemple de la bibliographie :

- a) modifier la feuille de style en XSLT de production en texte brut (page 63) afin de produire une liste des titres telle que :

```
Titres dans "bibliographie"
- XML cours et exercices
- Teach Yourself XML in 21 Days
- Normalisation de XML par le W3C
```

- b) retoucher la feuille de style précédente afin de numéroté la liste de titres :

```
Titres dans "bibliographie"
a) XML cours et exercices
b) Teach Yourself XML in 21 Days
c) Normalisation de XML par le W3C
```

indication : `<xsl:number count="motif" level="any" format="a" />`
qui produit le numéro (en lettre minuscule) du rang de l'élément courant parmi les éléments correspondant au motif de sélection indiqué et situés dans l'arborescence globale jusqu'à l'élément courant inclus.

- c) construire une feuille de style en XSLT pour produire une liste telle que :

```
<?xml version="1.0" encoding="UTF-8"?>
<liste-livre source="bibliographie">
  <livre id="Brillant 2007">XML cours et exercices</livre>
  <livre id="North Hermans 2000">Teach Yourself XML in 21 Days</livre>
</liste-livre>
```


EXERCICES SUR XSLT (SUITE)

2. Créer une feuille de style en XSLT pour transformer en XHTML le fil d'information en RSS vu précédemment (page 24), comme suit :

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:lang="fr" lang="fr">
<head>
<title>AgroParisTech - Actualité;</title>
</head>
<body>
<h1>AgroParisTech - Actualité;</h1>
<h2><a href="http://www.agroparistech.fr/+Conference+.html">
Conférence annuelle de l'Institut Européen des
Forêts (EFI) &grave; Nancy</a></h2>
<p>Cette année &#233;e &#224; l'occasion des 20 ans de l'EFI
(European Forest Institut), AgroParistech, l'Inra,
l'Université &#233; de Lorraine, et plusieurs autres partenaires
co-organisent une conf&#233;rence sur la forêt qui se
d&#233;roulera au Palais des Congr&#232;s de Nancy du 23 au 27
septembre 2013. Au programme, un forum sur les (...)</p>
</body>
</html>
```

qui correspond à :

AgroParisTech - Actualité

Conférence annuelle de l'Institut Européen des Forêts (EFI) à Nancy

Cette année à l'occasion des 20 ans de l'EFI (European Forest Institut), AgroParistech, l'Inra, l'Université de Lorraine, et plusieurs autres partenaires co-organisent une conférence sur la forêt qui se déroulera au Palais des Congrès de Nancy du 23 au 27 septembre 2013. Au programme, un forum sur les (...)

Indications

- Utiliser un attribut spécial de `<xsl:value-of>` pour annuler le recodage de symbole spéciaux (comme « & ») présents dans la section CDATA :
`<xsl:value-of disable-output-escaping="yes" select=" ... ">`
- Dans la balise `<html>`, retirer la déclaration de l'espace de nommage de XHTML (`xmlns="http://www.w3.org/1999/xhtml"`) car sinon il est produit des balises en HTML avec un attribut `xmlns` de valeur vide (`<p xmlns="">`)

ANALYSE D'UN DOCUMENT AVEC SAX

PRÉSENTATION

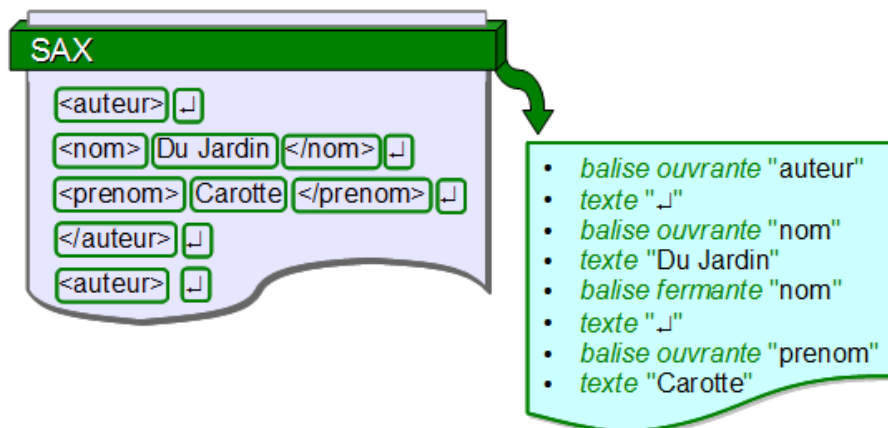
EVÉNEMENTS

EXEMPLE EN PYTHON

PRÉSENTATION DE SAX (*SIMPLE API FOR XML*)

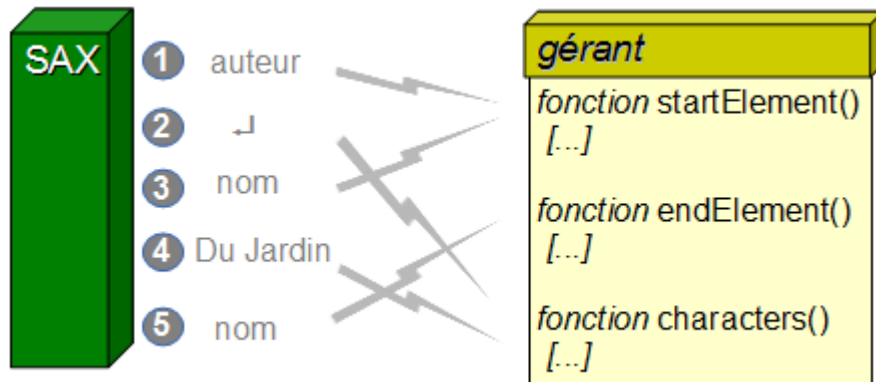
BIBLIOTHÈQUE D'ANALYSE LEXICALE D'UN DOCUMENT EN XML

• OBJECTIFS



- Analyse lexicale (*parsing*) d'un document en XML
reconnaissance de morceaux (*chunk*) : balise ¹ ou contenu (texte)
- Faciliter le développement du traitement d'un document en XML
bibliothèque dans le domaine public créée en 1998 (en Java initialement)

• PRINCIPALES CARACTÉRISTIQUES



- Modèle de programmation par événement
événement associé à chaque étape de l'analyse lexicale,
avec en correspondance une fonction dans le « gérant » ² associé
- Disponible dans de nombreux langages de programmation
Java, C++, Perl, PHP, Python, Javascript etc.
- Econome en utilisation de mémoire
pas de représentation complète du document en XML

UN OUTIL SIMPLE DEVENU UNE RÉFÉRENCE (<http://www.saxproject.org/>)

¹ A priori, les commentaires ne sont pas pris en compte (ils ne sont pas signalés par l'analyseur).

² Gérant (*handler*) : ensemble de fonctions prenant en charge chaque événement issu de l'analyse.

EVÉNEMENTS DÉFINIS DANS SAX

PRÉSENTATION MINIMALE DE SAX EN VERSION 2

- GESTION DES RÉSULTATS DE L'ANALYSE (CLASSE `contentHandler`)
 - `setDocumentLocator (source)`
fixation de la source de l'analyse
 - `startDocument ()`
début de l'analyse du document en XML
 - `startElement (nom, attributs)`
début d'un élément de XML avec son nom et ses attributs
 - `startElement (URI, nom local, nom qualifié, attributs)`
début d'un élément de XML avec espace de nommage identifié par URI, noms sans préfixe (local) et avec préfixe (qualifié), attributs
 - `endElement (nom)`
fin d'un élément de XML avec son nom
 - `endElement (URI, nom local, nom qualifié)`
fin d'un élément de XML avec espace de nommage identifié par URI, noms sans préfixe (local) et avec préfixe (qualifié)
 - `characters (texte)`
morceau de texte
 - `endDocument ()`
fin de l'analyse du document en XML
- GESTION DES ANOMALIES (CLASSE `errorHandler`)
 - `error (exception)`
signalement d'une anomalie n'empêchant pas la poursuite de l'analyse
 - `fatalError (exception)`
signalement d'une anomalie grave provoquant l'arrêt de l'analyse
 - `warning (exception)`
simple avertissement ne provoquant pas l'arrêt de l'analyse
 - `exception` dans la classe `SAXParseException` (fille de `SAXException`)
libellé de l'anomalie : `getMessage ()`
position dans le texte : `getColumnNumber ()` et `getLineNumber ()`
- CONTRÔLE DE L'ANALYSEUR LEXICAL (INTERFACE `XMLReader`)
 - `setContentHandler (ContentHandler gérant des résultats)`
fixation du gérant des résultats de l'analyse
 - `setErrorHandler (ErrorHandler gérant des anomalies)`
fixation du gérant des anomalies
 - `parse (InputSource document)`
démarrage de l'analyse

EXEMPLE EN PYTHON AVEC SAX

```
''' analyseur.py - Michel cartereau 22 IX 2013 - python 2.7
exemple simple d'analyse lexicale en XML avec SAX '''

import xml.sax

class GerantContenu(xml.sax.ContentHandler):

    def __init__(self):
        ''' report d'initialisation a la classe maternelle '''
        xml.sax.ContentHandler.__init__(self)

    def setDocumentLocator(self, locator):
        ''' informations sur le document a la source de l'analyse '''
        self.source = locator.getSystemId()

    def startDocument(self):
        ''' demarrage de l'analyse '''
        print "Analyse lexicale de '" + self.source + "'"
        self.niveau = 0

    def startElement(self, name, attrs):
        ''' arrivee d'une balise ouvrante d'element '''
        libelle = "Element '" + name
        for attribut in attrs.keys():
            libelle = libelle + " " + attribut + '=' + attrs[attribut] + " '
        print self.decalage() + libelle + " "
        self.niveau = self.niveau + 1

    def characters(self, content):
        ''' arrivee d'un contenu '''
        if not content.isspace():
            print self.decalage() + "Texte '" + content + "'"

    def endElement(self, name):
        ''' arrivee d'une balise fermante d'element '''
        self.niveau = self.niveau - 1
        print self.decalage() + "Fin de '" + name + "'"

    def endDocument(self):
        ''' fin de l'analyse '''
        print "Fin."

    def decalage(self):
        ''' renvoi d'un texte de decalage selon 'niveau' '''
        resultat = '';
        for i in range(self.niveau):
            resultat = resultat + "| "
        return resultat + " "
```

avec le démarrage de l'analyse par :

```
xml.sax.parse( nom du fichier1, GerantContenu() )
```

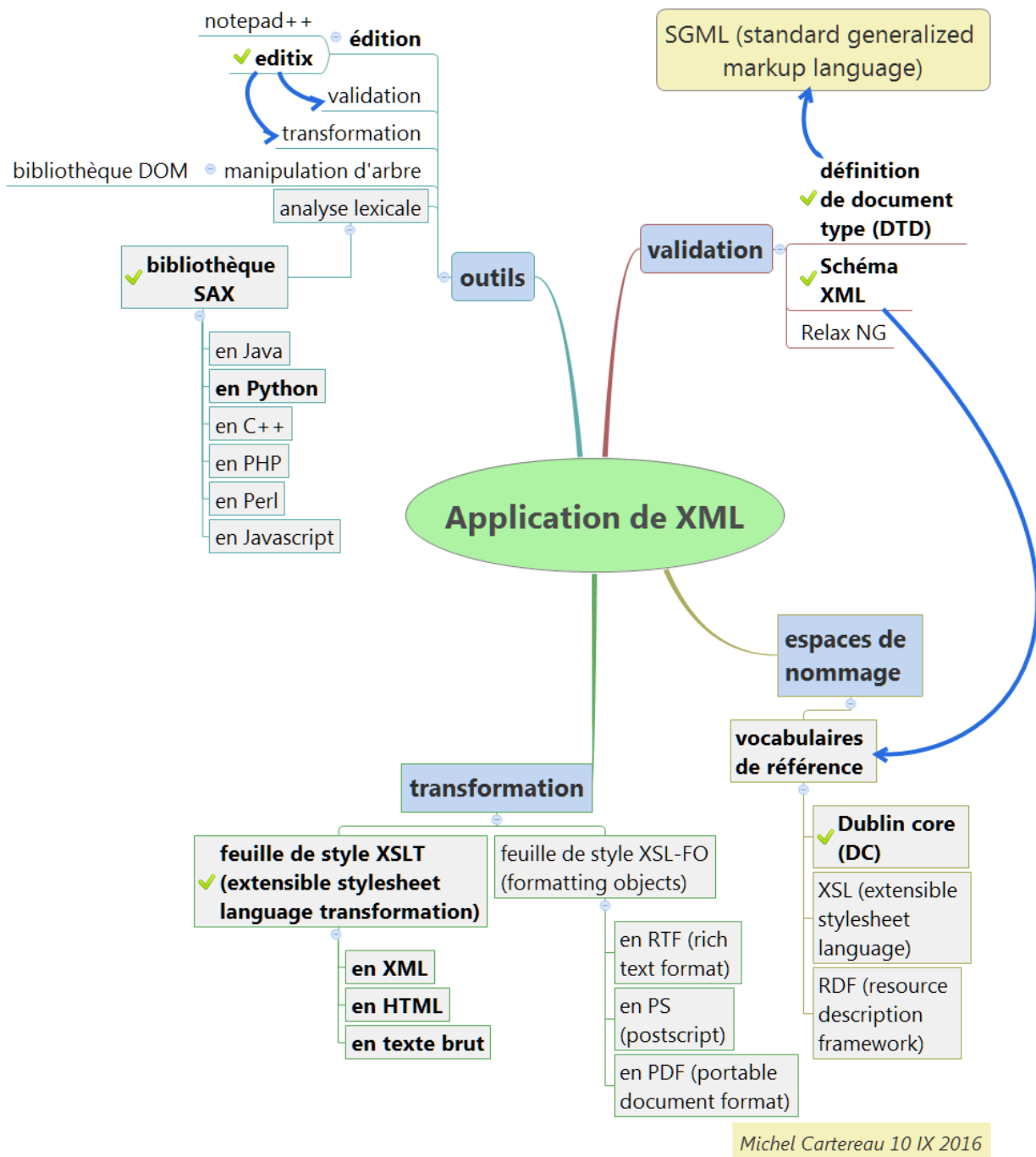
¹ Si le document en XML indique un fichier de schéma (<!DOCTYPE>), ce fichier doit être accessible par l'analyseur ; on peut désactiver son utilisation via l'attribut `standalone="yes"` dans la balise `<?xml>`

EXEMPLE EN PYTHON AVEC SAX (SUITE)

```
# exemple d'analyse
xml.sax.parse("bibliographie.xml", GerantContenu() )

Analyse lexicale de 'bibliographie.xml'
Element 'bibliographie'
| Element 'reference type-ref="livre" id-ref="Brillant 2007" '
| | Element 'titre'
| | | Texte 'XML cours et exercices'
| | Fin de 'titre'
| | Element 'auteur'
| | | Element 'nom'
| | | | Texte 'Brillant'
| | | Fin de 'nom'
| | | Element 'prenom'
| | | | Texte 'Alexandre'
| | | Fin de 'prenom'
| | Fin de 'auteur'
| | Element 'annee'
| | | Texte '2007'
| | Fin de 'annee'
| Fin de 'reference'
| Element 'reference type-ref="livre" id-ref="North Hermans 2000" '
| | Element 'titre'
| | | Texte 'Teach Yourself XML in 21 Days'
| | Fin de 'titre'
| | Element 'auteur'
| | | Element 'nom'
| | | | Texte 'North'
| | | Fin de 'nom'
| | | Element 'prenom'
| | | | Texte 'Simon'
| | | Fin de 'prenom'
| | Fin de 'auteur'
| | Element 'auteur'
| | | Element 'nom'
| | | | Texte 'Hermans'
| | | Fin de 'nom'
| | | Element 'prenom'
| | | | Texte 'Paul'
| | | Fin de 'prenom'
| | Fin de 'auteur'
| | Element 'annee'
| | | Texte '2000'
| | Fin de 'annee'
| Fin de 'reference'
| Element 'reference type-ref="site" id-ref="W3C XML" '
| | Element 'titre'
| | | Texte 'Normalisation de XML par le W3C'
| | Fin de 'titre'
| | Element 'adresse'
| | | Texte 'http://www.w3.org/standards/xml/'
| | Fin de 'adresse'
| Fin de 'reference'
Fin de 'bibliographie'
Fin.
```

RÉCAPITULATIF



ANNEXES

RÉCAPITULATIF DES PROPRIÉTÉS DE CSS

VOCABULAIRE DE DUBLIN CORE

GRAMMAIRES DE DTD, XML ET XPATH

ALPHABETS INFORMATIQUES : ASCII, ISO 8859, UNICODE

RÉCAPITULATIF DE PROPRIÉTÉS DE CSS

PROPRIÉTÉS ÉLÉMENTAIRES DE CSS (*CASCADING STYLE SHEETS*)

MISE EN FORME DE TEXTE

`color:` (couleur du texte)

nom ou codage de couleur

`font:`

raccourci pour une taille suivie de(s) police(s), le tout éventuellement précédé dans l'ordre par l'italique et-ou les petites majuscules et-ou la graisse

`font-family:` (polices)

un ou une liste de nom de police (séparée par une virgule)

`font-style:` (italique)

normal **ou** italic **ou** oblique

`font-size:` (taille)

xx-small **ou** x-small **ou** small **ou** medium **ou** large **ou** x-large **ou** xx-large **ou** larger **ou** smaller **ou** une taille absolue **ou** un pourcentage

`font-variant:` (petites majuscules)

normal **ou** small-caps

`font-weight:` (graisse)

normal **ou** bold **ou** bolder **ou** lighter **ou** 100 **ou** 200 ... **ou** 900

`letter-spacing:` (espacement des lettres)

normal **ou** une taille absolue **ou** un pourcentage

`text-decoration:` (décoration)

underline (souligné) **ou** overline (surligné) **ou** line-through (rayé) **ou**

blink (clignotant)

`text-transform:` (casse)

uppercase (majuscules) **ou** lowercase (minuscules)

ou capitalize (initiales des mots en majuscule) **ou** none

`word-spacing:` (espacement des mots)

normal **ou** une taille

`white-space:` (traitement des espaces dans le texte)

normal **ou** pre (respect des espaces multiples et des changements de ligne) **ou**

nowrap (pas de césure)

LISTES

`list-style:`

raccourci pour une ou plusieurs des propriétés de liste décrites ci-après

`list-style-image:` (image en symbole de liste)

adresse d'image **ou** none

`list-style-position:` (position du symbole de liste)

outside (avec retrait) **ou** inside (sans retrait)

`list-style-type:` (type du symbole de liste)

disc (rond) **ou** circle (cercle) **ou** decimal (chiffres arabes) **ou** lower-roman (chiffres romains minuscules) **ou** upper-roman (chiffres romains majuscules) **ou** lower-alpha (lettres minuscules) **ou** upper-alpha (lettres majuscules) **ou** none

MISE EN FORME D'UN BLOC

`text-indent`: (décalage en début de paragraphe)
une mesure ou un pourcentage

`line-height`: (interligne)
normal ou un pourcentage ou une mesure ou un nombre

`text-align`: (alignement horizontal)
left (gauche) ou right (droite) ou center (milieu) ou justify (justification à droite)

`vertical-align`: (alignement vertical)
sub (indice) ou super (exposant) ou baseline (première ligne) ou top (haut) ou text-top (haut de la première ligne) ou middle (milieu) ou bottom (bas) ou text-bottom (bas de la dernière ligne) ou pourcentage

`width`: (largeur d'une image ou d'un bloc de texte)
auto ou une mesure ou un pourcentage

`height`: (hauteur d'une image ou d'un bloc de texte)
auto ou une mesure

`float`: (incrustation)
left (à gauche) ou right (à droite) ou none

`clear`: (passage sous une incrustation)
left (à gauche) ou right (à droite) ou both (des deux côtés) ou none

`display`: (changement de niveau lors de l'affichage)
block (bloc) ou inline (élément de ligne) ou list-item (élément de liste) ou none

ARRIÈRE-PLAN

`background`:
raccourci pour une ou plusieurs des propriétés d'arrière-plan décrites ci-après

`background-attachment`: (défilement de l'image de fond)
fixed (fixe) ou scroll (défilement)

`background-color`: (couleur de fond)
une couleur ou transparent

`background-image`: (image de fond)
adresse d'une image ou none

`background-position`: (position initiale de l'image de fond)
une valeur (horizontal avec vertical à 50 %) ou deux valeurs (horizontal puis vertical),
en vertical : mesure ou top (haut) ou center (milieu) ou bottom (bas),
en horizontal : mesure ou left (gauche) ou center (milieu) ou right (droite)

`background-repeat`: (répétition de l'image de fond en mosaïque)
repeat (en largeur et en hauteur) ou repeat-x (en largeur) ou repeat-y (en hauteur) ou no-repeat (aucune répétition)

BORDURES

`border`: **ou** `border-left`: **ou** `border-right`: **ou** `border-top`: **ou** `border-bottom`:
raccourcis pour indiquer la couleur et-ou le type de trait et-ou l'épaisseur,
soit pour les 4 bordures à la fois, soit pour une bordure particulière

`border-color`: (couleur des bordures)

1 couleur (commune aux 4 côtés) ou 2 (haut-bas puis droite-gauche) ou 3 (haut puis droite-gauche puis bas) ou 4 (haut puis droite puis bas puis gauche)

`border-style`: (type de trait des bordures)

1 valeur (commune aux 4 côtés) ou 2 (haut-bas puis droite-gauche) ou 3 (haut puis droite-gauche puis bas) ou 4 (haut puis droite puis bas puis gauche) :

`none` **ou** `dotted` (pointillé) **ou** `dashed` (tirets) **ou** `solid` (continu) **ou** `double` (doublé) **ou** `groove` (gravé) **ou** `ridge` (en arête) **ou** `inset` (en retrait) **ou** `outset` (en avant)

`border-width`: (épaisseur des bordures)

1 valeur (commune aux 4 côtés) ou 2 (haut-bas puis droite-gauche) ou 3 (haut puis droite-gauche puis bas) ou 4 (haut puis droite puis bas puis gauche) :

`thin` (fine) **ou** `medium` (moyenne) **ou** `thick` (épaisse) **ou** une mesure

`border-left-width`: **ou** `border-right-width`: **ou**

`border-bottom-width`: **ou** `border-top-width`: (épaisseur respective d'une des bordures)

`thin` (fine) **ou** `medium` (moyenne) **ou** `thick` (épaisse) **ou** une mesure

MARGES

`margin`: (tailles des marges externes)

1 valeur (commune aux 4 côtés) ou 2 (haut-bas puis droite-gauche) ou 3 (haut puis droite-gauche puis bas) ou 4 (haut puis droite puis bas puis gauche) :

`auto` **ou** une mesure **ou** un pourcentage

`margin-left`: **ou** `margin-right`: **ou** `margin-top`: **ou** `margin-bottom`:
(taille d'une marge externe sur l'un des côtés)

`auto` **ou** une mesure **ou** un pourcentage

`padding`: (tailles des marges internes)

1 valeur (commune aux 4 côtés) ou 2 (haut-bas puis droite-gauche) ou 3 (haut puis droite-gauche puis bas) ou 4 (haut puis droite puis bas puis gauche) :

une mesure **ou** un pourcentage

`padding-left`: **ou** `padding-right`: **ou** `padding-top`: **ou** `padding-bottom`:
(taille d'une marge interne sur l'un des côtés)

une mesure **ou** un pourcentage

NOTATIONS DE VALEURS

couleur

soit un nom : `aqua` `black` `blue` `fuchsia` `gray` `green` `lime` `maroon` `navy` `olive` `purple` `red` `silver` `teal` `yellow` `white`

soit un code : `#RRVBB` **ou** `rgb(rouge%, vert%, bleu%)`

taille

soit absolue (unités) : `in` (pouce), `cm` (centimètre), `mm` (millimètre), `pt` (point typographique), `pc` (pica, 12 points), `px` (pixel), `em` (hauteur moyenne de lettre), `ex` (largeur moyenne de lettre)

soit relative : `%` (pourcentage)

(attention : pas de blanc inséré entre la valeur et le symbole d'unité ou %)

VOCABULAIRE DE DUBLIN CORE (DC)

15 éléments initialement définis en 1995 et repris dans la norme ISO 15836

Espace de nommage associé : <http://purl.org/dc/terms/>
(préfixe `dc:` par convention)

Elément	Rôle
<code>title</code>	Titre principal du document
<code>creator</code>	Nom de la personne, de l'organisation ou du service à l'origine de la rédaction du document
<code>subject</code>	Mots-clefs, phrases de résumé, ou codes de classement
<code>description</code>	Résumé, table des matières, ou texte libre. Raffinements : table des matières, résumé
<code>publisher</code>	Nom de la personne, de l'organisation ou du service à l'origine de la publication du document
<code>contributor</code>	Nom d'une personne, d'une organisation ou d'un service qui contribue ou a contribué à l'élaboration du document. Chaque contributeur fait l'objet d'un élément Contributor séparé
<code>date</code>	Date d'un évènement dans le cycle de vie du document
<code>type</code>	Genre du contenu
<code>format</code>	Type MIME (RFC 2046), ou format physique du document
<code>identifier</code>	Identificateur non ambigu : il est recommandé d'utiliser un système de référencement précis, afin que l'identifiant soit unique au sein du site, par exemple les URI ou les numéros ISBN.
<code>source</code>	Ressource dont dérive le document : le document peut découler en totalité ou en partie de la ressource en question. Il est recommandé d'utiliser une dénomination formelle des ressources(exemple : URI)
<code>language</code>	Langue de la ressource (codification recommandée : RFC 4646)
<code>relation</code>	Lien avec d'autres ressources
<code>coverage</code>	Couverture spatiale (point géographique, pays, régions, noms de lieux) ou temporelle
<code>rights</code>	Droits de propriété intellectuelle, Copyright, droits de propriété divers

A noter : ce vocabulaire originel est désormais remplacé par sa forme étendue à 55 « termes » (et non plus « éléments ») , référencié par l'espace de nommage <http://purl.org/dc/terms/> et le préfixe `dct:` (ou `dcterms:`)

RÉFÉRENCES : <http://dublincore.org/documents/dcmi-terms/>

http://fr.wikipedia.org/wiki/Dublin_Core

EXEMPLES DE GRAMMAIRES D'ÉCRITURE

EXEMPLE DE L'ÉCRITURE D'UN NOM D'ÉLÉMENT OU D'ATTRIBUT

• NOTATION EN EBNF (VERSION DU W3C)

▪ Principales notations

`::=` définition d'un composant

`' '` ou `" "` texte final

`|` ou (alternative)

`[c1-c2]` un caractère dans l'intervalle de c₁ à c₂

`()` regroupement

`*` zéro ou plusieurs répétitions de ce qui précède

▪ Exemple

```
Nom ::= Caractere_debut_nom(Caractere_nom) *
```

```
Caractere_debut_nom ::= ":" | [A-Z] | "_" | [a-z]
```

```
Caractere_nom ::= Caractere_debut_nom | "-" | "." | [0-9]
```

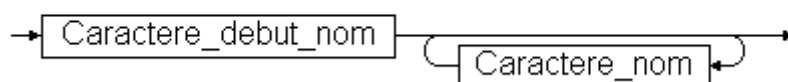
• NOTATION EN DIAGRAMMES SYNTAXIQUES

▪ Notation à l'aide de graphiques ¹

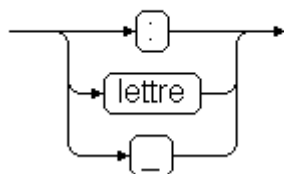
définition d'un composant comme des chemins possibles,
avec notation de composant dans un rectangle à bords droits
et d'un texte dans un rectangle à bords arrondis (ou un cercle)

▪ Exemple

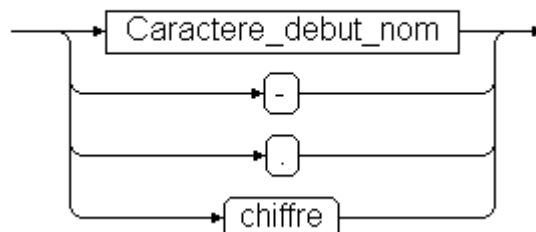
Nom



Caractere_debut_nom



Caractere_nom

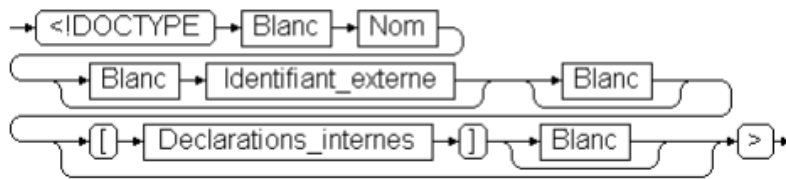


NOTATIONS FORMELLES INDÉPENDANTES DE XML

¹ Appelé en anglais « *railroad diagram* » ; outil de construction en ligne à <http://railroad.my28msec.com/>

GRAMMAIRE DE DTD

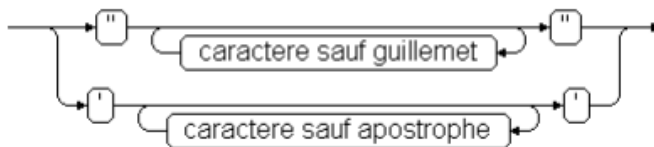
Declaration_type_document



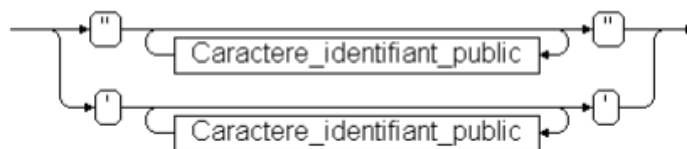
Identifiant_externe



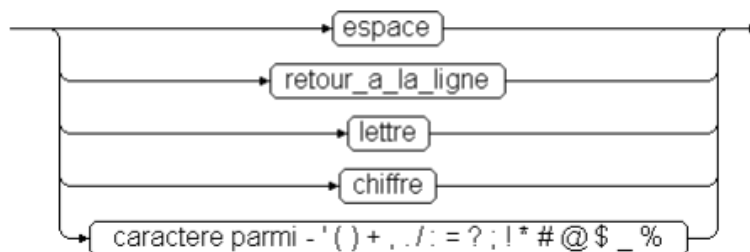
Adresse



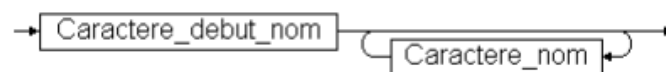
Valeur_identifiant_public



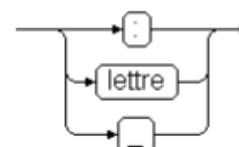
Caractere_identifiant_public



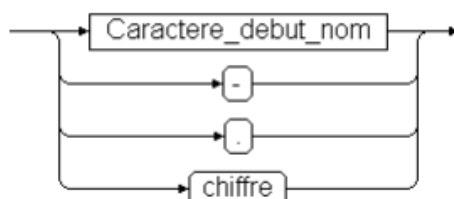
Nom



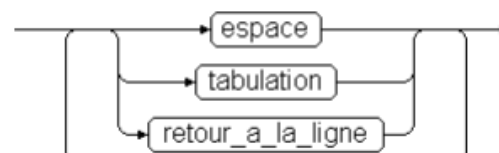
Caractere_debut_nom



Caractere_nom

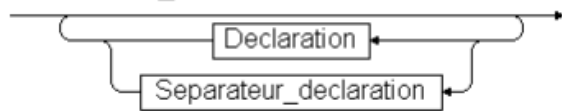


Blanc

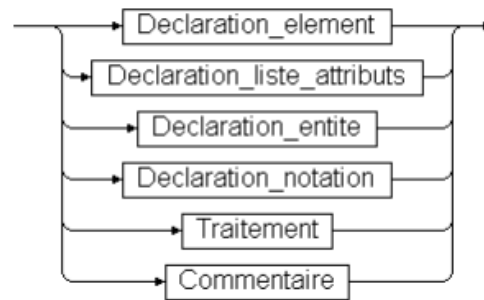


GRAMMAIRE DE DTD (SUITE)

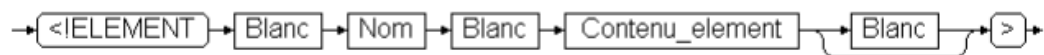
Declarations_internes



Declaration



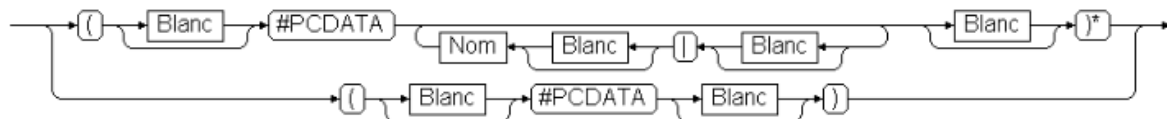
Declaration_element



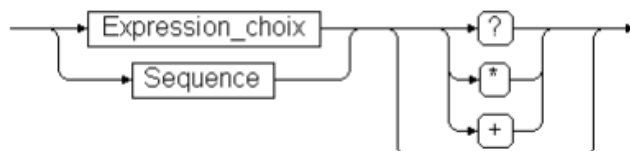
Contenu_element



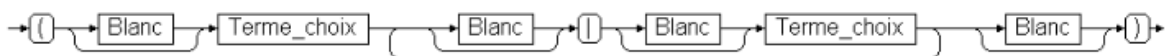
Contenu_melange



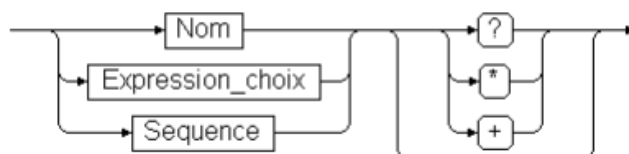
Contenu_enfants



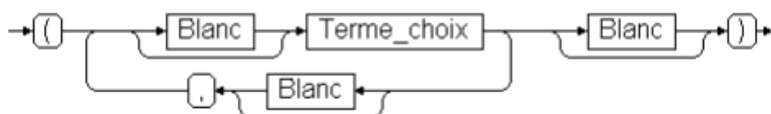
Expression_choix



Terme_choix

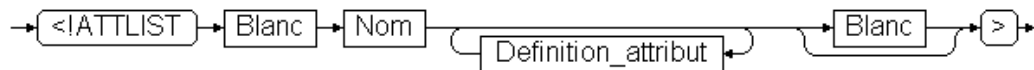


Sequence

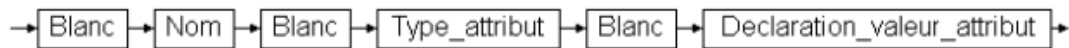


GRAMMAIRE DE DTD (SUITE)

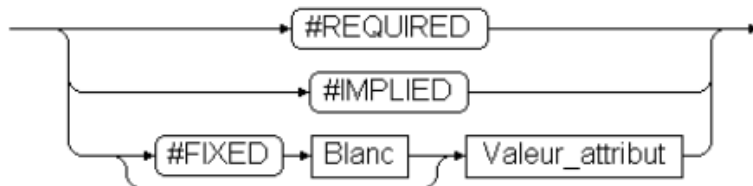
Declaration_liste_attributs



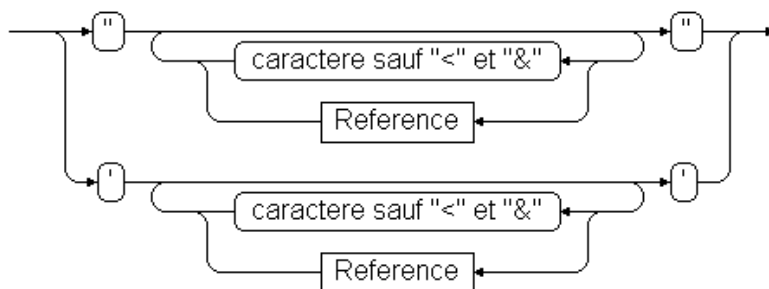
Definition_attribut



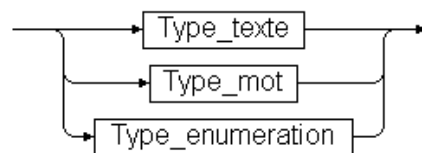
Declaration_valeur_attribut



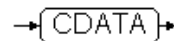
Valeur_attribut



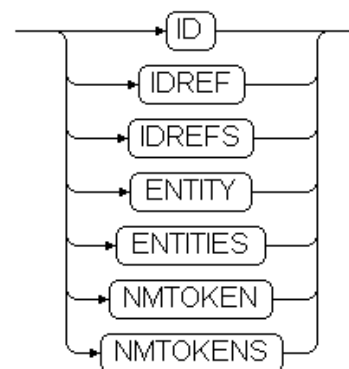
Type_attribut



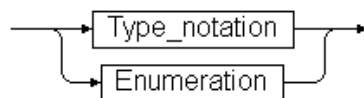
Type_texte



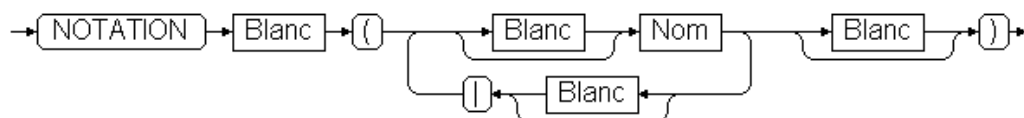
Type_mot



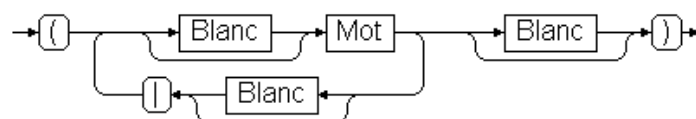
Type_enumeration



Type_notation

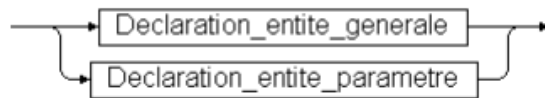


Enumeration

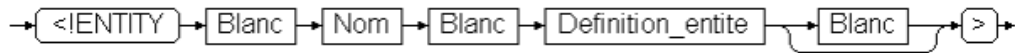


GRAMMAIRE DE DTD (SUITE)

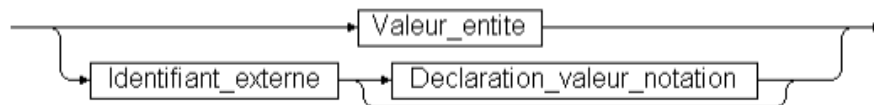
Declaration_entite



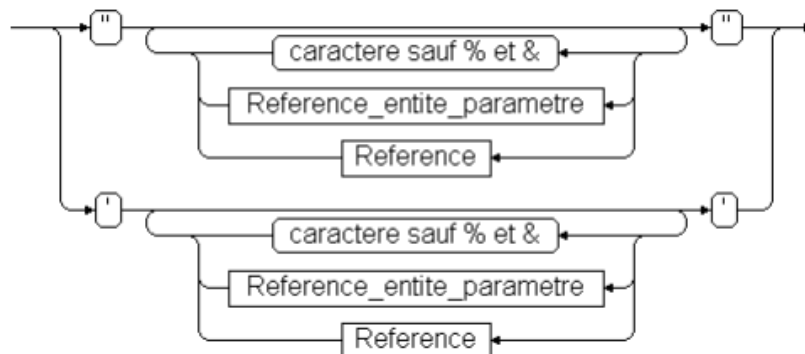
Declaration_entite_generale



Definition_entite



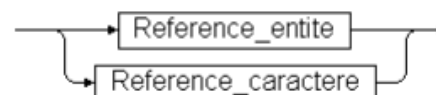
Valeur_entite



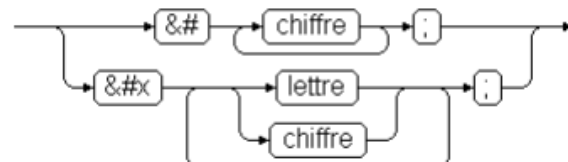
Reference_entite_parametre



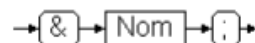
Reference



Reference_caractere



Reference_entite

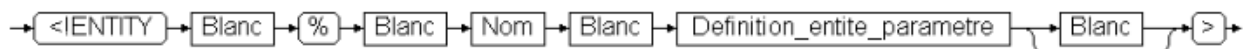


Declaration_valeur_notation

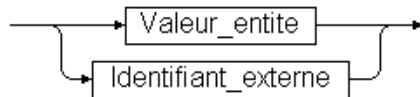


GRAMMAIRE DE DTD (FIN)

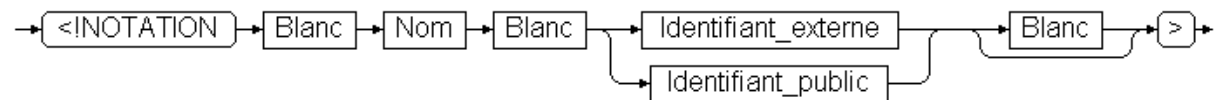
Declaration_entite_parametre



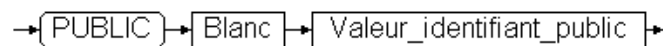
Definition_entite_parametre



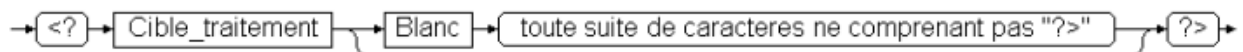
Declaration_notation



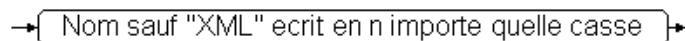
Identifiant_public



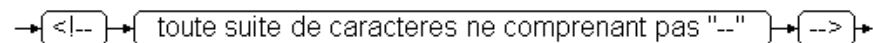
Traitement



Cible_traitement



Commentaire



Separateur_declaration



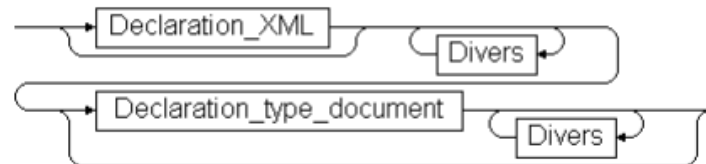
GRAMMAIRE DE XML

Voir aussi la grammaire précédente de DTD pour les éléments non détaillés.

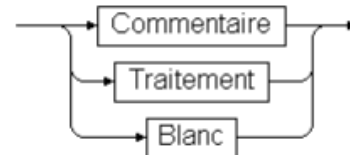
Document



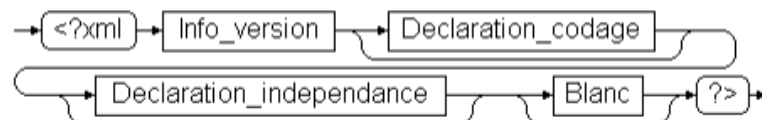
Prologue



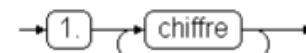
Divers



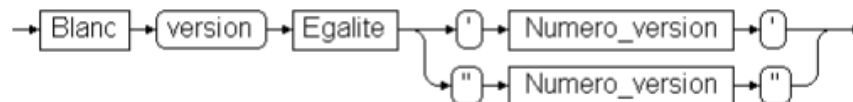
Declaration_XML



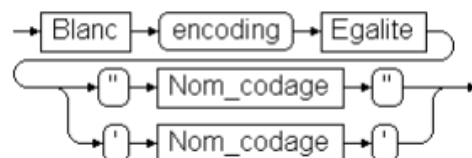
Numero_version



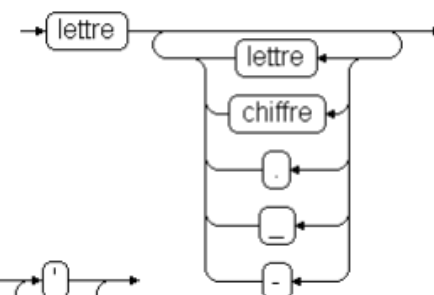
Info_version



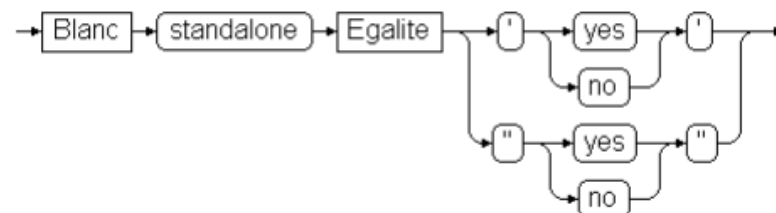
Declaration_codage



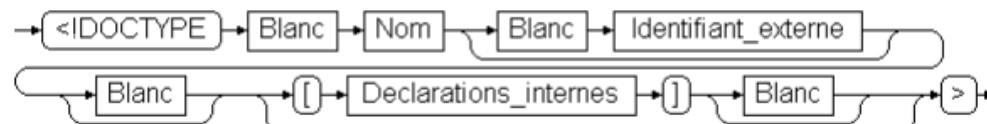
Nom_codage



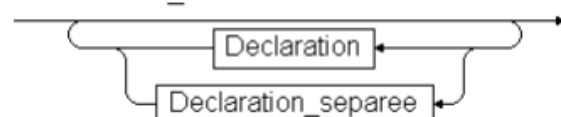
Declaration_independance



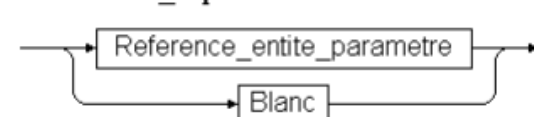
Declaration_type_document



Declarations_interne

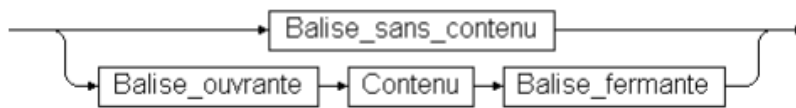


Declaration_separee

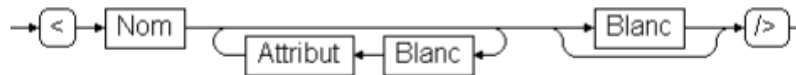


GRAMMAIRE DE XML (FIN)

Element



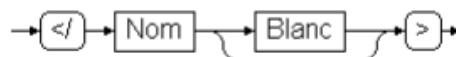
Balise_sans_contenu



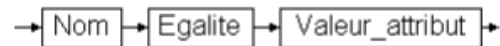
Balise_ouvrante



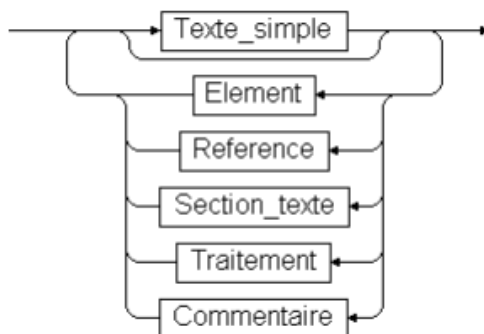
Balise_fermante



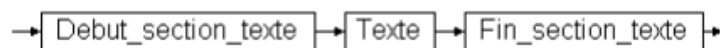
Attribut



Contenu



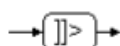
Section_texte



Debut_section_texte



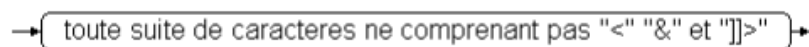
Fin_section_texte



Texte



Texte_simple

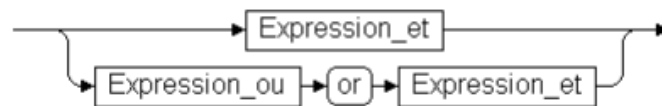


GRAMMAIRE DE XPATH

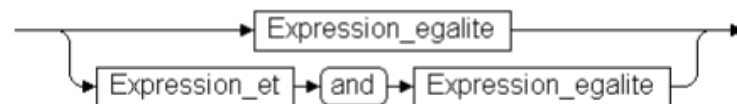
Expression

→ Expression_ou

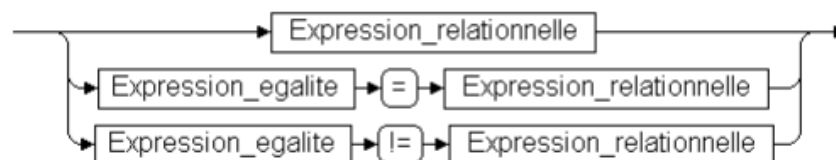
Expression_ou



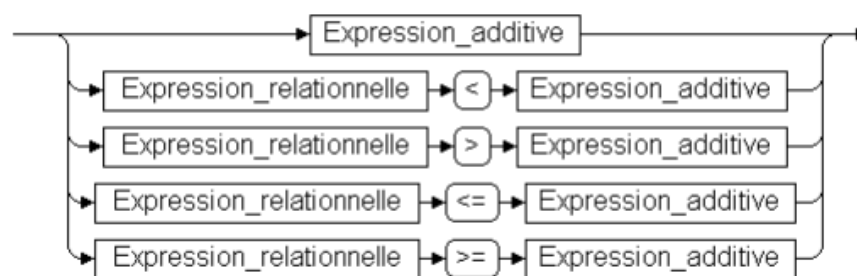
Expression_et



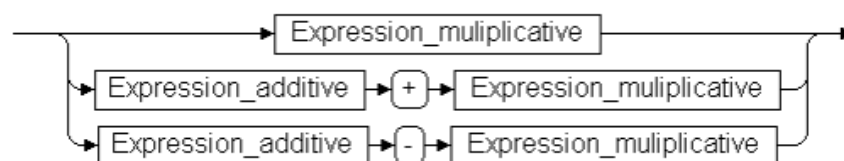
Expression_egalite



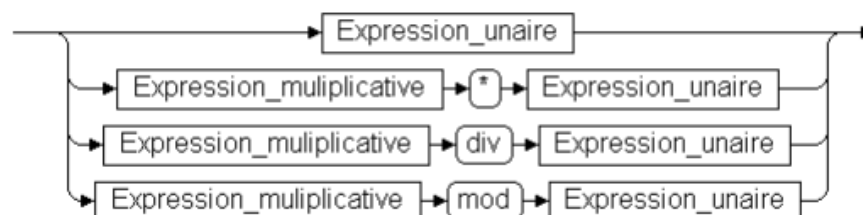
Expression_relationnelle



Expression_additive

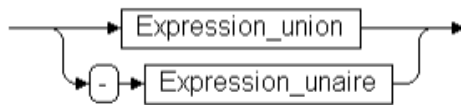


Expression_multiplicative

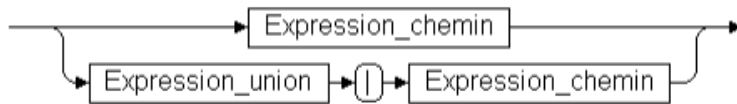


GRAMMAIRE DE XPATH (SUITE)

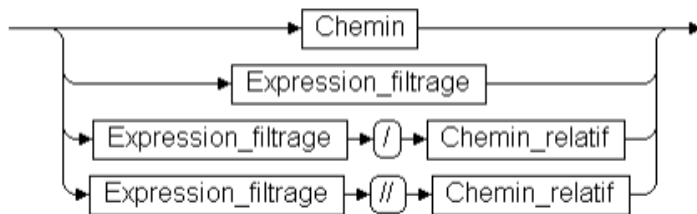
Expression_unaire



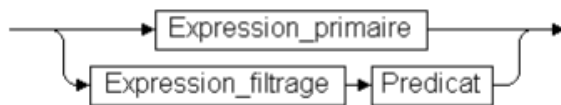
Expression_union



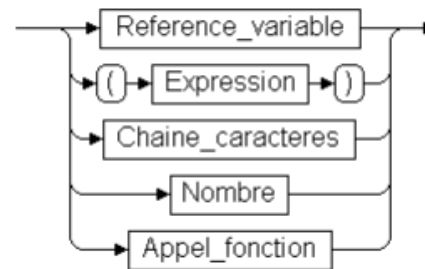
Expression_chemin



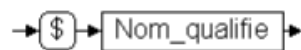
Expression_filtirage



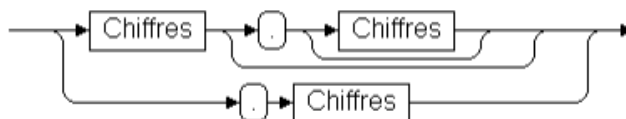
Expression_primaire



Reference_variable



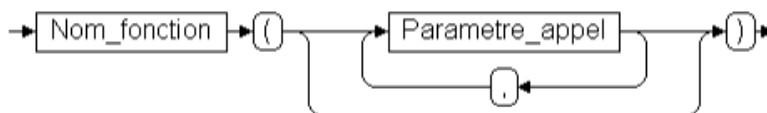
Nombre



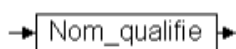
Chiffres



Appel_fonction



Nom_fonction

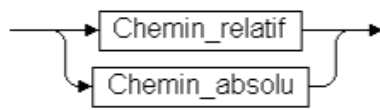


Parametre_appel

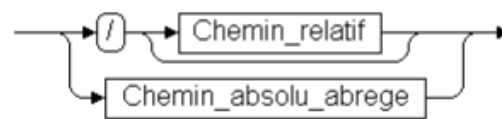


GRAMMAIRE DE XPATH (SUITE)

Chemin



Chemin_absolu



Chemin_absolu_abrege



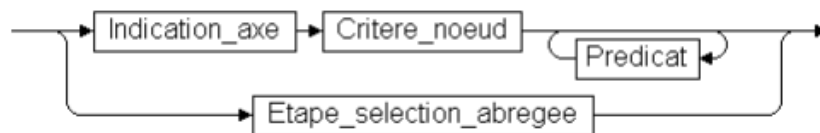
Chemin_relatif



Chemin_relatif_abrege



Etape_selection



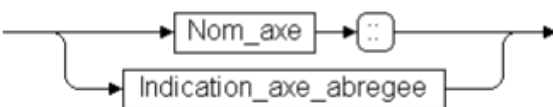
Predicat



Expression_predicat



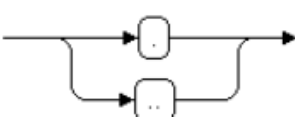
Indication_axe



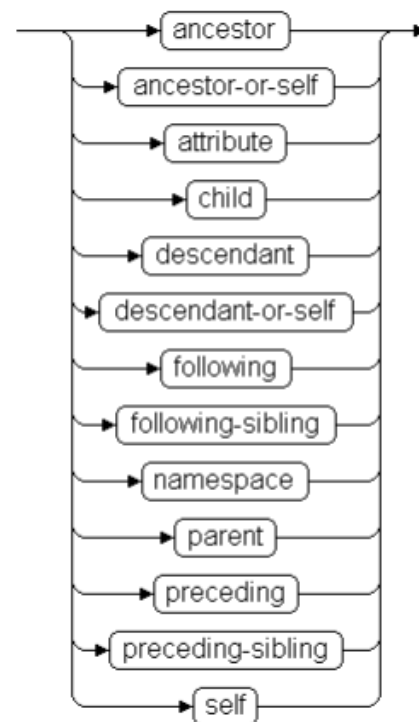
Indication_axe_abregee



Etape_selection_abregee

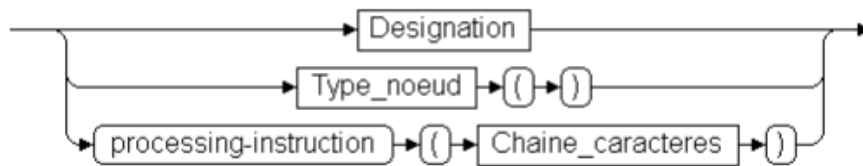


Nom_axe

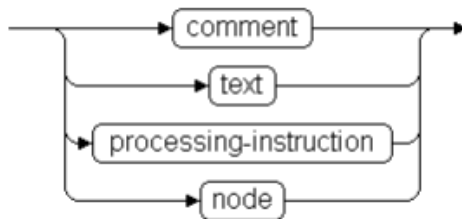


GRAMMAIRE DE XPATH (FIN)

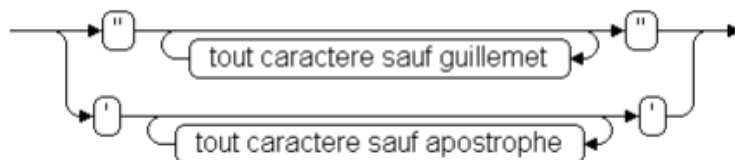
Critere_noeud



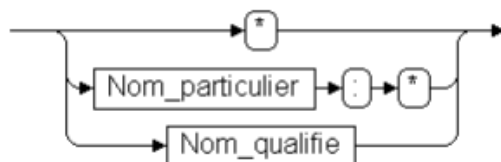
Type_noeud



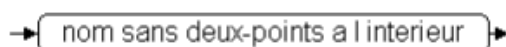
Chaine_caracteres



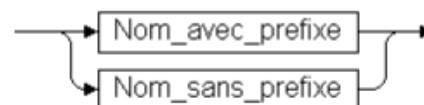
Designation



Nom_particulier



Nom_qualifie



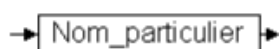
Nom_avec_prefixe



Nom_sans_prefixe



Prefixe



Nom_local



ALPHABETS INFORMATIQUES

- CODE ASCII (*AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE*)

- Première norme, définie aux Etats-unis (1968)
code sur 7 bits, jeu de caractères anglo-saxons (US-ASCII)
- Normalisation internationale : ISO 646 (1972)
avec transformation de caractères par pays, ex : France

#	@	[\]	{		}
£	à	°	ç	§	é	ù	è

- Adaptations et extensions multiples
ASCII étendu à 8 bits (Macintosh, MS-Dos ou Windows)

- CODE EBCDIC (*EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE*)

- Code spécifique aux grands systèmes d'IBM (≈ 1965)
code sur 8 bits, incompatible avec l'ASCII

- CODE NORMALISÉ SUR 8 BITS : ISO 8859 (1987)

- Découpage de la plage de codage en 4 zones
C0 (CL) 00h-1Fh et G0 (GL) 20h-7Eh : US-ASCII
C1 (CR) 80h-9Fh : inutilisé, G1 (GR) A0h-FEh : variable
- Variantes par zones géographiques (G1)
Europe de l'Ouest (1), de l'Est (2), du Sud (3), du Nord (4)
cyrillique (5), arabe (6), grec (7), hébreu (8), etc.

- UNICODE (1991)

- Code à vocation universelle : alphabets, idéogrammes, etc.
compatible avec ISO/IEC 10646 : UCS (*universal character set*)
- Plage de codage sur 32 bits, découpée en zones
encodage normalisé sur 1, 2 ou 4 octets (*little/big endian*)
UTF-8, 16 LE/BE, 32 LE/BE (*Unicode transformation format*)
- Définition par un consortium, en lien avec l'OSI
initiative de Apple et Xerox

CODE ASCII NORMALISÉ (US-ASCII)

CODE SUR 7 BITS

0	00h	Ctrl+@	NULL	32	20h	SP	64	40h	@	96	60h	`
1	01h	Ctrl+A	SOH	33	21h	!	65	41h	A	97	61h	a
2	02h	Ctrl+B	STX	34	22h	"	66	42h	B	98	62h	b
3	03h	Ctrl+C	ETX	35	23h	#	67	43h	C	99	63h	c
4	04h	Ctrl+D	EOT	36	24h	\$	68	44h	D	100	64h	d
5	05h	Ctrl+E	ENQ	37	25h	%	69	45h	E	101	65h	e
6	06h	Ctrl+F	ACK	38	26h	&	70	46h	F	102	66h	f
7	07h	Ctrl+G	BEL	39	27h	'	71	47h	G	103	67h	g
8	08h	Ctrl+H	BS	40	28h	(72	48h	H	104	68h	h
9	09h	Ctrl+I	HT	41	29h)	73	49h	I	105	69h	i
10	0Ah	Ctrl+J	LF	42	2Ah	*	74	4Ah	J	106	6Ah	j
11	0Bh	Ctrl+K	VT	43	2Bh	+	75	4Bh	K	107	6Bh	k
12	0Ch	Ctrl+L	FF	44	2Ch	,	76	4Ch	L	108	6Ch	l
13	0Dh	Ctrl+M	CR	45	2Dh	-	77	4Dh	M	109	6Dh	m
14	0Eh	Ctrl+N	SO	46	2Eh	.	78	4Eh	N	110	6Eh	n
15	0Fh	Ctrl+O	SI	47	2Fh	/	79	4Fh	O	111	6Fh	o
16	10h	Ctrl+P	DLE	48	30h	0	80	50h	P	112	70h	p
17	11h	Ctrl+Q	DC1	49	31h	1	81	51h	Q	113	71h	q
18	12h	Ctrl+R	DC2	50	32h	2	82	52h	R	114	72h	r
19	13h	Ctrl+S	DC3	51	33h	3	83	53h	S	115	73h	s
20	14h	Ctrl+T	DC4	52	34h	4	84	54h	T	116	74h	t
21	15h	Ctrl+U	NAK	53	35h	5	85	55h	U	117	75h	u
22	16h	Ctrl+V	SYN	54	36h	6	86	56h	V	118	76h	v
23	17h	Ctrl+W	ETB	55	37h	7	87	57h	W	119	77h	w
24	18h	Ctrl+X	CAN	56	38h	8	88	58h	X	120	78h	x
25	19h	Ctrl+Y	EM	57	39h	9	89	59h	Y	121	79h	y
26	1Ah	Ctrl+Z	SUB	58	3Ah	:	90	5Ah	Z	122	7Ah	z
27	1Bh	Ctrl+[ESC	59	3Bh	;	91	5Bh	[123	7Bh	{
28	1Ch	Ctrl+\	FS	60	3Ch	<	92	5Ch	\	124	7Ch	
29	1Dh	Ctrl+]	GS	61	3Dh	=	93	5Dh]	125	7Dh	}
30	1Eh	Ctrl+^	RS	62	3Eh	>	94	5Eh	^	126	7Eh	~
31	1Fh	Ctrl+_	US	63	3Fh	?	95	5Fh	_	127	7Fh	DEL

SYMBOLES DE FONCTION DU CODE ASCII

déc hex Nom	terme anglais	terme français
0 00 NULL	<i>Null</i>	Nul
1 01 SOH	<i>Start of heading</i>	Début d'entête
2 02 STX	<i>Start of text</i>	Début de texte
3 03 ETX	<i>End of text</i>	Fin de texte
4 04 EOT	<i>End of transmit</i>	Fin de communication
5 05 ENQ	<i>Enquiry</i>	Demande
6 06 ACK	<i>Acknowledge</i>	Accusé de réception
7 07 BEL	<i>Bell</i>	Sonnerie
8 08 BS	<i>Backspace</i>	Retour arrière
9 09 HT	<i>Horizontal tab</i>	Tabulation horizontale
10 0A LF	<i>Line feed</i>	Interligne
11 0B VT	<i>Vertical tab</i>	Tabulation verticale
12 0C FF	<i>Form feed</i>	Page suivante
13 0D CR	<i>Carriage return</i>	Retour en début de ligne
14 0E SO	<i>Shift out</i>	Hors code
15 0F SI	<i>Shift in</i>	En code
16 10 DLE	<i>Data line escape</i>	Echappement en transmission
17 11 DC1	<i>Device control 1</i>	Commande auxiliaire n° 1
18 12 DC2	<i>Device control 2</i>	Commande auxiliaire n° 2
19 13 DC3	<i>Device control 3</i>	Commande auxiliaire n° 3
20 14 DC4	<i>Device control 4</i>	Commande auxiliaire n° 4
21 15 NAK	<i>Negative acknowledge</i>	Accusé de réception négatif
22 16 SYN	<i>Synchronous idle</i>	Synchronisation
23 17 ETB	<i>End of transmit block</i>	Fin de bloc transmis
24 18 CAN	<i>Cancel</i>	Annulation
25 19 EM	<i>End of medium</i>	Fin de support
26 1A SUB	<i>Substitute</i>	Remplacement
27 1B ESC	<i>Escape</i>	Echappement
28 1C FS	<i>File separator</i>	Séparateur de fichier
29 1D GS	<i>Group separator</i>	Séparateur de groupe
30 1E RS	<i>Record separator</i>	Séparateur d'enregistrement
31 1F US	<i>Unit separator</i>	Séparateur d'unité
127 7F DEL	<i>Delete</i>	Effacement

CODE ISO 8859

VERSION LATIN-1 (EUROPE OCCIDENTALE) : ISO 8859-1

déc	hex		déc	hex		déc	hex	
160	A0	no-break space	192	C0	À	224	E0	à
161	A1	í	193	C1	Á	225	E1	á
162	A2	ç	194	C2	Â	226	E2	â
163	A3	£	195	C3	Ã	227	E3	ã
164	A4	¤	196	C4	Ä	228	E4	ä
165	A5	¥	197	C5	Å	229	E5	å
166	A6		198	C6	Æ	230	E6	æ
167	A7	§	199	C7	Ç	231	E7	ç
168	A8	¨	200	C8	È	232	E8	è
169	A9	©	201	C9	É	233	E9	é
170	AA	ª	202	CA	Ê	234	EA	ê
171	AB	«	203	CB	Ë	235	EB	ë
172	AC	¬	204	CC	Ì	236	EC	ì
173	AD	-	205	CD	Í	237	ED	í
174	AE	®	206	CE	Î	238	EE	î
175	AF	¯	207	CF	Ï	239	EF	ï
176	B0	°	208	D0	Ð	240	F0	ð
177	B1	±	209	D1	Ñ	241	F1	ñ
178	B2	²	210	D2	Ò	242	F2	ò
179	B3	³	211	D3	Ó	243	F3	ó
180	B4	´	212	D4	Ô	244	F4	ô
181	B5	µ	213	D5	Õ	245	F5	õ
182	B6	¶	214	D6	Ö	246	F6	ö
183	B7	·	215	D7	×	247	F7	÷
184	B8	,	216	D8	Ø	248	F8	ø
185	B9	¹	217	D9	Ù	249	F9	ù
186	BA	º	218	DA	Ú	250	FA	ú
187	BB	»	219	DB	Û	251	FB	û
188	BC	¼	220	DC	Ü	252	FC	ü
189	BD	½	221	DD	Ý	253	FD	ý
190	BE	¾	222	DE	Þ	254	FE	þ
191	BF	¿	223	DF	ß	255	FF	ÿ

RÉFÉRENCE : <http://alis.isoc.org/codage/iso8859/jeuxiso.htm>



Notation : UH₁H₂H₃H₄
pour la partie débutant au rang
noté H₁H₂H₃H₄ en hexadécimal.

(ISO 8859-1)

U0000	C0 Controls and Basic Latin
U0080	C1 Controls and Latin-1 Supplement

U0100	Latin Extended-A
U0180	Latin Extended-B
U0250	IPA Extensions
U02B0	Spacing Modifier Letters
U0300	Combining Diacritical Marks
U0370	Greek
U0400	Cyrillic
U0530	Armenian
U0590	Hebrew
U0600	Arabic
U0900	Devanagari
U0980	Bengali
U0A00	Gurmukhi
U0A80	Gujarati
U0B00	Oriya
U0B80	Tamil
U0C00	Telugu
U0C80	Kannada
U0D00	Malayalam
U0E00	Thai
U0E80	Lao
U0F00	Tibetan
U10A0	Georgian
U1100	Hangul Jamo
U1200	Ethiopic
U13A0	Cherokee
U1400	Unified Canadian Aboriginal Syllabic
U1680	Ogham
U16A0	Runic
U1780	Khmer
U1800	Mongolian

U1E00	Latin Extended Additional
U1F00	Greek Extended

U2000	General Punctuation
U2070	Superscripts and Subscripts
U20A0	Currency Symbols
U20D0	Combining Diacritical Marks for Symbols
U2100	Letterlike Symbols
U2150	Number Forms
U2190	Arrows
U2200	Mathematical Operators
U2300	Miscellaneous Technical
U2400	Control Pictures
U2440	Optical Character Recognition
U2460	Enclosed Alphanumerics
U2500	Box Drawing
U2580	Block Elements
U25A0	Geometric Shapes
U2600	Miscellaneous Symbols
U2700	Dingbats
U2800	Braille Patterns
U2E80	CJK Radicals Supplemen
U2F00	Kangxi Radicals
U2FF0	Ideographic Description Characters
U3000	CJK Symbols and Punctuation
U3040	Hiragana
U30A0	Katakana
U3100	Bopomofo
U3130	Hangul Compatibility Jamo
U3190	Kanbun
U21A0	Bopomofo Extended
U3200	Enclosed CJK Letters and Months
U3300	CJK Compatibility
U3400	CJK Unified Ideographs Extension A

U4E00	CJK Unified Ideographs
-------	------------------------

UNICODE™ (SUITE)

UA000	Yi Syllables
UA490	Yi Radicals
UAC00	Hangul Syllables
UD800	High Surrogates
UDB80	High Private Use Surrogates
UDC00	Low Surrogates
UE000	Private Use Area
UF900	CJK Compatibility Ideographs
UFB00	Alphabetic Presentation Forms
UFB50	Arabic Presentation Forms-A
UFE20	Combining Half Marks
UFE30	CJK Compatibility Forms
UFE50	Small Form Variants
UFE70	Arabic Presentation Forms-B
UFF00	Halfwidth and Fullwidth Forms
UFFF0	Specials

U10000	Linear B Syllabary
U10080	Linear B Ideograms
U10100	Aegean Numbers
U10300	Old Italic
U10330	Gothic
U10380	Ugaritic
U10400	Deseret
U10450	Shavian
U10480	Osmanya
U10800	Cypriot Syllabary
U1D000	Byzantine Musical Symbols
U1D100	Musical Symbols
U1D300	Tai Xuan Jing Symbols
U1D400	Mathematical Alphanumeric Symbols
U20000	CJK Unified Ideographs Extension B (13MB)
U2F800	CJK Compatibility Ideographs Supplement
UE0000	Tags
UE0100	Variation Selectors Supplement
UF0000	Supplementary Private Use Area-A

U100000	Supplementary Private Use Area-B
---------	----------------------------------

RÉFÉRENCE : <http://www.unicode.org/charts/>

UNICODE SUR UN OCTET

UTF-8 : *UNICODE TRANSFORMATION FORMAT - 8 BITS*

● PRINCIPE

Encodage par octet

un symbole d'UNICODE codé par une séquence de 1 à 6 octets

Séquence d'1 seul octet codée avec le bit de poids fort à 0

codage équivalent à US-ASCII pour les 128 premières valeurs

Séquence de plusieurs (N) octets avec le bit de poids fort à 1

1^{er} octet : n bits de poids forts à 1, suivis par un bit à zéro

octets suivants : bits de poids forts à 10

bits restants remplis par le rang en UNICODE, à partir du plus faible

Table de correspondance

Unicode (hexadécimal)	UTF-8 (binaire)
0000 0000 à 0000 007F	0 xxxxxxx
0000 0080 à 0000 07FF	110 xxxxx 10 xxxxxx
0000 0800 à 0000 0FFF	1110 xxxx 10 xxxxxx 10 xxxxxx
0001 0000 à 001F FFFF	11110 xxx 10 xxxxxx 10 xxxxxx 10 xxxxxx
0020 0000 à 03FF FFFF	111110 xx 10 xxxxxx 10 xxxxxx 10 xxxxxx 10 xxxxxx
0400 0000 à 7FFF FFFF	1111110 x 10 xxxxxx 10 xxxxxx 10 xxxxxx 10 xxxxxx 10 xxxxxx

● EXEMPLES

X : U0058 (00000000 01011000 h)

1 octet en UTF-8 : **0**1011000 h

© : U00A9 (00000000 10101001 h)

2 octets en UTF-8 : **110**00010 h **10**101001 h

RÉFÉRENCE : RFC 2279, <http://www.ietf.org/rfc/rfc2279.txt>

INDEX

- , 40]	<xsl:when>, 57
' , 40, 57]]>, 16	=
! , 40	{	=, 40
!=, 40	{ }, 56	>
" , 40		>, 40
#	, 15, 38	A
#FIXED, 15	+	abréviation, 16
#IMPLIED, 15	+, 15, 40	alphabet, 89
#PCDATA, 15	<	analyse lexicale, 67
#REQUIRED, 15	<, 13, 40	ancestor, 39
&	<!-- -->, 7, 16	ancestor-or-self, 39
&, 13, 57	<![CDATA[, 16	and, 40
", 6	<![ATTLIST>, 15	ANY, 15
', 6	<![DOCTYPE>, 17	application XML, 21
&#code;, 16	<![ELEMENT>, 15	arbre, 5
ode;, 16	<![ENTITY>, 16	ASCII, 89
&;, 16	<?xml>, 7	Atom, 3
', 6	<?xml-stylesheet>, 9, 61	attribut, 6
", 6	<=, 40	attribut, valeur par défaut, 15
*	<xs:any>, 32	attribute, 39
* , 15, 39, 40	<xs:attribute>, 30	B
,	<xs:choice>, 32	background:, 74
., 15	<xs:complexType>, 30, 31	background-attachment:, 74
.	<xs:element>, 28, 30, 31, 32	background-color:, 74
.., 39	<xs:extension>, 31	background-image:, 74
.xsd, 28	<xs:restriction>, 31	background-position:, 74
/	<xs:schema>, 28	background-repeat:, 74
/, 38	<xs:sequence>, 32	balise à contenu, 6
//, 38	<xs:simpleContent>, 31	BE, 89
?	<xsl:apply-templates>, 54, 58	blanc, 16, 29, 38, 56
?, 15	<xsl:attribute>, 59	bonne écriture, 13
@	<xsl:attribute-set>, 59	boolean(), 40, 43
@, 40	<xsl:choose>, 57	border:, 75
[<xsl:element>, 59	border-bottom:, 75
[], 40	<xsl:for-each>, 58	border-bottom-width:, 75
	<xsl:if>, 57	border-color:, 75
	<xsl:otherwise>, 57	border-left:, 75
	<xsl:output />, 52	border-left-width:, 75
	<xsl:sort>, 58	border-right:, 75
	<xsl:stylesheet>, 52	border-right-width:, 75
	<xsl:template>, 52, 53	border-style:, 75
	<xsl:text>, 56, 63	border-top:, 75
	<xsl:transform>, 52	border-top-width:, 75
	<xsl:value-of>, 56	border-width:, 75
	<xsl:variable>, 57	

<p>C</p> <p>CDATA, 13, 15, 16</p> <p>child, 39</p> <p>clear:, 74</p> <p>codage, 7</p> <p>cohérence, 13</p> <p>color:, 73</p> <p>comment(), 39</p> <p>commentaire, 7, 16</p> <p>concat(), 41</p> <p>contains(), 41</p> <p>conversion, 41, 43</p> <p>count(), 41</p> <p>CSS, 3, 9</p> <p>CSS1, 73</p>	<p>font-size:, 73</p> <p>font-style:, 73</p> <p>font-variant:, 73</p> <p>font-weight:, 73</p> <p>G</p> <p>gérant, 67</p> <p>H</p> <p>handler, 67</p> <p>height:, 74</p> <p>HTML, 50</p>	<p>N</p> <p>name(), 41, 45</p> <p>namespace, 59</p> <p> :, 45</p> <p>namespace-uri(), 45</p> <p>NaN, 41</p> <p>navigateur, 8</p> <p>NMTOKEN, 15</p> <p>NMTOKENS, 15</p> <p>node(), 39</p> <p>nœud, 42</p> <p>noeud d'attribut, 35, 39</p> <p>noeud d'élément, 35</p> <p>nom complet, 22</p> <p>nom qualifié, 22</p> <p>normalize-space(), 41</p> <p>not(), 40</p> <p>Notepad++, 10</p> <p>number(), 41, 43</p>
<p>D</p> <p>descendant, 39</p> <p>descendant-or-self, 39</p> <p>diagramme syntaxique, 77</p> <p>disable-output-escaping, 65</p> <p>display:, 74</p> <p>div, 40</p> <p>document bien écrit, 13</p> <p>document bien formulé, 13</p> <p>document valide, 13</p> <p>DOM, 3</p> <p>DTD, 3, 13, 25</p> <p>DTD, Grammaire, 78</p> <p>Dublin core, 76</p>	<p>I</p> <p>ID, 15</p> <p>id(), 41</p> <p>IDREF, 15</p> <p>IDREFS, 15</p> <p>indent, 52</p> <p>ISO 10646, 89</p> <p>ISO 646, 89</p> <p>ISO 8601, 29</p> <p>ISO 8859, 89, 92</p> <p>J</p> <p>Jaxe, 10</p> <p>L</p> <p>last(), 41</p> <p>Latin-1, 92</p> <p>LE, 89</p> <p>letter-spacing:, 73</p> <p>line-height:, 74</p> <p>list-style:, 73</p> <p>list-style-image:, 73</p> <p>list-style-position:, 73</p> <p>list-style-type:, 73</p> <p>local-name(), 45</p>	<p>O</p> <p>OpenDocument, 3</p> <p>OpenXML, 3</p> <p>optional, 30</p> <p>or, 40</p> <p>ordre des éléments, 39</p> <p>OWL, 3</p>
<p>E</p> <p>EBCDIC, 89</p> <p>EBNF, 13, 77</p> <p>Editix, 10</p> <p>élément, 5</p> <p>EMPTY, 15</p> <p>encoding, 7, 52</p> <p>ensemble de nœuds, 42</p> <p>entité, 16</p> <p>espace, 29, 38, 56</p> <p>espace de nommage, 22, 45</p>	<p>M</p> <p>margin:, 75</p> <p>margin-bottom:, 75</p> <p>margin-left:, 75</p> <p>margin-right:, 75</p> <p>margin-top:, 75</p> <p>match, 53</p> <p>MathML, 3</p> <p>maxOccurs, 32</p> <p>method, 52</p> <p>minOccurs, 32</p> <p>mixed, 32</p> <p>mod, 40</p>	<p>P</p> <p>padding:, 75</p> <p>padding-bottom:, 75</p> <p>padding-left:, 75</p> <p>padding-right:, 75</p> <p>padding-top:, 75</p> <p>parent, 39</p> <p>parsing, 67</p> <p>path, 35, 36</p> <p>PCDATA, 16</p> <p>position(), 41</p> <p>preceding, 39</p> <p>preceding-sibling, 39</p> <p>prédicat, 42</p> <p>préfixe, 22</p> <p>processeur, 50</p> <p>processing-instruction(), 39</p> <p>prologue, 7, 17</p> <p>propriété, 73</p>
<p>F</p> <p>false(), 40</p> <p>faux, 40, 41</p> <p>feuille de style, 9</p> <p>float:, 74</p> <p>following, 39</p> <p>following-sibling, 39</p> <p>font:, 73</p> <p>font-family:, 73</p>	<p>R</p> <p>racine, 35</p> <p>rang de nœud, 42</p> <p>RDF, 3</p>	

Relax NG, 13
required, 30
retour à la ligne, 38
retour-à-la-ligne, 29, 56
round(), 41
RSS, 3, 24

S

saut-de-ligne, 29, 56
SAX, 67
SAX, événements, 68
SAX, Python, 69
schéma de structuration, 13
Schéma XML, 3, 13, 28
attribut, 30, 31
cardinalité, 32
choix d'éléments, 32
élément à contenu
complexe, 32
élément à contenu mixte, 32
élément à contenu simple,
31
Élément à contenu vide, 30
facette, 29
liste d'éléments, 32
regroupement, 32
restriction, 29
type complexe, 28
type dérivé, 29
type mixte, 28
type simple, 28
types de données, 29
select, 54
SGML, 3, 13
SMIL, 3
standalone, 69
starts-with(), 41
string(), 41, 43
string-length(), 41
substring(), 41
sum(), 41
SVG, 3
SYSTEM, 16, 17

T

tabulation, 15, 16, 29, 38, 41,
56
text(), 39
text-align:, 74

text-indent:, 74
text-transform:, 73
true(), 40
type de document, 13
type dérivé, 29

U

UCS, 89
unbounded, 32
UNICODE, 89, 93, 95
URI, 22, 45
US-ASCII, 89, 90
use-attribute-sets, 59
UTF-8, 89, 95

V

vérification, 13
version, 7
vertical-align:, 74
vocabulaire, 13
vrai, 41

W

W3C, 3
white-space:, 73
width:, 74
word-spacing:, 73

X

XML, grammaire, 83
XMLmind XML Editor, 10
xmlns, 25
xmlns:, 22
XPath, 3, 35, 50, 54, 56
axe, 36, 39
booléen, 40
chaîne de caractères, 40
chemin, 35, 36, 38
chemin absolu, 38
chemin relatif, 38
contexte, 36
conversion, 40
critère, 36, 39
décimal, 40
entier, 40
étape, 36, 38
expression, 35, 38
filtre, 36
fonction, 41

formule, 38, 40
grammaire, 85
logique, 40
noeud courant, 38
noeud de texte, 39
prédicat, 36, 40
texte d'élément, 40
union, 38
valeur d'attribut, 40
variable, 40
XQuery, 3
xs:, 28
xs:boolean, 29
xs:date, 29
xs:dateTime, 29
xs:double, 29
xs:duration, 29
xs:float, 29
xs:hexBinary, 29
xs:integer, 29
xs:normalizedString, 29
xs:string, 29
xs:time, 29
xs:token, 29
xsl:, 52
XSL-FO, 50
XSLT, 3, 50
attribut, 59
CDATA, 65
choix, 57
élément, 59
espace, 56
espace de nommage, 65
feuille de style, 50
itération, 58
jeu d'attributs, 59
modèle, 50, 53, 54
saut de ligne, 63
stylesheet, 50
template, 50
texte, 56, 63
tri, 58
valeur d'attribut, 56
variable, 57
XML, 62
XSLT-FO, 3

SOMMAIRE

PRÉSENTATION GÉNÉRALE	2
Qu'est-ce que XML ?	3
Exemple de données : bibliographie	4
Structuration en XML	5
Balisage en XML	6
Document en XML	7
Visualisation directe dans un navigateur	8
Visualisation avec une feuille de style	9
Exemple d'éditeur spécialisé : EditiX	10
Exercices sur la structuration en XML	11
TYPE DE DOCUMENT (DTD)	12
Généralités sur la vérification	13
Modèle en DTD (<i>document type definition</i>)	14
Notation de DTD (<i>document type definition</i>)	15
Notation d'une DTD : compléments	16
Indication d'une DTD pour le document	17
Manipulation de DTD avec EditiX	18
Exercices sur les DTD	19
ESPACES DE NOMMAGE	20
Cas d'ambiguïté	21
Principes de l'espace de nommage	22
Exemple de la bibliographie localisée	23
Exemple d'un fil d'information (RSS)	24
DTD et espace de nommage	25
Exercices sur les espaces de nommage	26
SCHÉMA XML	27
Présentation de Schéma XML	28
Schéma XML : types de données	29
Schéma XML : attribut, élément à contenu vide	30
Schéma XML : élément à contenu simple	31
Schéma XML : élément à contenu complexe, mixte	32
Exemple de Schema XML pour la bibliographie	33
SÉLECTION DANS UN DOCUMENT AVEC XPATH	34
Présentation de XPath (<i>XML path language</i>)	35
Expression de XPath : généralités	36
Expression de XPath : exemples introductifs	37
Expression de XPath : forme générale	38
Expression de XPath : axe et critère	39
Expression de XPath : formule et prédicat	40
Expression de XPath : fonctions	41
Expression de XPath : nœuds et évaluation	42
Expression de XPath : conversions	43
Expression de XPath : exemples avancés	44
Expression de XPath : espace de nommage	45
Manipulation de XPath avec EditiX	46
XPath et espaces de nommages avec EditiX	47
Exercices sur les expressions de XPath	48

TRANSFORMATION D'UN DOCUMENT AVEC XSLT	49
Présentation de XSLT	50
XSLT : Exemple de la bibliographie	51
Langage XSLT : squelette	52
Langage XSLT : modèle de transformation	53
Langage XSLT : modèle de transformation (bis)	54
Langage XSLT : algorithme de transformation	55
Langage XSLT : production de texte	56
Langage XSLT : variable, choix.....	57
Langage XSLT : itération, tri.....	58
Langage XSLT : création d'élément, d'attribut.....	59
Feuille de style en XSLT de l'exemple.....	60
Transformation en XSLT avec EditiX	61
Exemple de transformation en XML	62
Exemple de transformation en texte brut.....	63
Exercices sur XSLT	64
Exercices sur XSLT (suite)	65
ANALYSE D'UN DOCUMENT AVEC SAX	66
Présentation de SAX (<i>Simple API for XML</i>).....	67
Événements définis dans SAX	68
Exemple en Python avec SAX.....	69
Exemple en Python avec SAX (suite).....	70
Récapitulatif.....	71
ANNEXES.....	72
Récapitulatif de propriétés de CSS	73
Vocabulaire de Dublin core (DC).....	76
Exemples de grammaires d'écriture	77
Grammaire de DTD	78
Grammaire de DTD (suite)	79
Grammaire de DTD (suite)	80
Grammaire de DTD (suite)	81
Grammaire de DTD (fin)	82
Grammaire de XML	83
Grammaire de XML (fin)	84
Grammaire de XPath.....	85
Grammaire de XPath (suite).....	86
Grammaire de XPath (suite).....	87
Grammaire de XPath (fin).....	88
Alphabets informatiques	89
Code ASCII normalisé (US-ASCII)	90
Symboles de fonction du code ASCII	91
Code ISO 8859.....	92
UNICODE™	93
UNICODE™ (suite)	94
UNICODE sur un octet	95
INDEX.....	96