Langages et outils du Web

Pr. Sidi Mohammed BENSLIMANE

ÉCOLE SUPÉRIEURE EN INFORMATIQUE 08 MAI 1945

– SIDI BEL ABBES –

s.benslimane@esi-sba.dz

Objectifs et Prérequis du cours

OBJECTIFS
L'objectif du cours est d'aider l'élève à: - Apprendre la syntaxe XML et les langages (outils) sous-jacents. - Étudier et mettre en œuvre les langages (outils) pour la manipulation (programmation) de XML : production, exploitation, diffusion et stockage de données XML.

PRÉREQUIS	Algorithmique, Structures de données, Programmation Orientée objet et Technologie Web

Contenu du programme

- XML: Origine et Concepts de base
- Les Grammaires XML
  - ✓ Validation de document XML: DTD, XML Schéma.
- Recherche et transformation de documents XML
  - ✓ XPATH, XSLT.
- Interrogation de documents XML
  - ✓ Xquery, XQUF.
- Programmation XML
  - ✓ DOM, SAX.
- Quelques applications de XML
  - ✓ Les services Web

XML: Origine et Concepts de base

1. ORIGINES DE XML
2. HTML AVANTAGES ET INCONVÉNIENTS
3. LES OBJECTIFS DE XML
4. LA STRUCTURE DE XML
5. LES ATOUTS DE XML
6. LES APPLICATIONS DE XML
6. LES ÉDITEURS DE XML

XML : Origines

- **1986: SGML** (Standard Generalized Markup Language) a été créé pour mémoriser, de manière structurée, de grosses documentations techniques (avions, armée, etc.) ou lexicographiques (dictionnaires). Il devient une norme de l'*ISO* (ISO-8879).

- **1990** : **HTML** (*Hypertext Markup Language*) est un sous-ensemble très spécialisé de SGML conçu pour représenter les pages web.

- **1996**: **XML** (eXtensible Markup Language) (Langage de marquage extensible ou langage à balises étendu) est un Standard open source développé par XML Working Group sous l'égide de **Jon Bosak** *du* World Wide Web Consortium (W3C).

- **1998**: **XML 1.0** est recommandée officiellement par le (W3C)

World Wide Web Consortium (http://www.w3.org)

- W3C est un consortium fondé par Tim Berners-Lee en 1994 pour promouvoir la compatibilité des technologies du web.
- Le W3C n'émet pas des normes, mais des recommandations
- La gestion du W3C est assurée conjointement par:
  1. Le Massachusetts Institute of Technology aux USA,
  2. L'ERCIM (European Research Consortium for Informatics and Mathematics) en Europe,
  3. L'Université Keio au Japon.
- 434 membres industriels en Aout 2020
  - Google, Amazon, Apple, Dell, IBM, Facebook, Intel, Microsoft, Cisco, eBay, Nokia, Sony, etc,

Un document est composé de 4 éléments

Émetteur

Destinataire

Date

Référence

Contenu

LA STRUCTURE

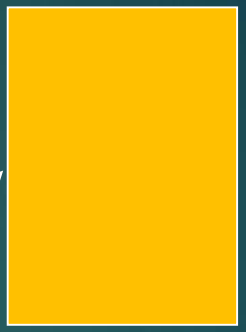
De : Chef Service Budget

À : M. le PDG

Le 11/02/2020

NOTE n° 132-A

La Direction du budget a évalué les besoins en credits de paiement pour l'année 2020 à 100 Millions DA.....



LE SUPPORT PHYSIQUE

Chef Service Budget

M. Le PDG

11/02/2020

132-A

La Direction du Budget....

LES DONNEES

Alignement

Couleur

Fonte

LA MISE EN FORME

Un document numérique est composé de 3 éléments

DOCUMENT NUMERIQUE = STRUCTURE + DONNEES + MISE EN FORME

1 FICHIER pour la structure (.xsd ou .dtd)

1 FICHIER pour les données (.xml)

1 FICHIER pour la mise en forme (.xsl ou .css)

HTML : présentation

- Proposé par le W3C comme format de documents sur le Web en 1990.
- Langage avec des balises fixes standardisées permettant la mise en forme d'un texte.
- Standard reconnu par tous les navigateurs, très populaire sur le Web.
- Nouvelle version HTML 5.3, Janvier 2021.

```
<HTML>
<HEAD>
<TITLE> Titre du document </TITLE>
</HEAD>
<BODY>
<H1> Entete du document</H1>
<p> Contenu du document </p>
</BODY>
</HTML>
```

HTML : inconvénients

- HTML a un ensemble pré-déterminé de balises
  - Il n'est pas possible de créer de nouvelles balises.
- HTML est un langage de présentation
  - Les balises donnent des indications sur la manière de présenter et non sur le contenu.
- Mise à jour d'un ensemble de pages difficile :
  - Restructuration ou remise en forme de l'ensemble des pages du site fastidieux.
- Pas de rigueur d'écriture:
  - On peut écrire indifféremment une balise en minuscules ou en majuscules,
  - ne pas indiquer la balise de fermeture
  - faire chevaucher des balises différentes.

XML: définitions de base

- XML est un méta-langage universel qui permet de standardiser la manière dont l'information est :
  - Structurée (DTD, XML-SCHEMA)
  - Présentée (XSL, CSS)
  - Transformée (XSLT)
  - Retrouvée (XPATH, XQUERY)
  - Archivée (XML-DataBases)
  - Cryptée (XML-Encryption)
  - Échangée (SOAP)
  - Annotée (RDF, OWL)
- XML est Supporté par les grands constructeurs:
  - IBM, HP, SUN, Intel, Sony, Dell, Microsoft, etc.
- Deux concepts fondamentaux
  1. Structure, contenu et présentation sont séparés
  2. Les balises ne sont pas figées

Les objectifs de conception

- Les documents XML doivent être lisibles par l'homme et raisonnablement clairs.
- XML doit pouvoir être utilisé sans difficulté sur Internet.
- XML doit soutenir une grande variété d'applications.
- XML doit être compatible avec HTML.
- Il doit être facile de créer des documents XML.
- Il doit être facile d'écrire des programmes traitant les documents XML.

Structure d'un document XML

Un document XML est constitué de 4 éléments :

- Un Prologue(facultatif),
- Un élément racine (et un seul),
- Un arbre d'éléments et éventuellement leurs attributs,
- Des Commentaires (facultatifs).

Structure d'un document XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- ceci est une carte de visite -->
<carteDeVisite>
  <prénom>Sidi Mohamed</prénom>
  <nom>BENSLIMANE</nom>
  <adresse>
    <numéro>42</numéro>
    <voie type="avenue">1er Novembre 1954</voie>
    <ville codepostal="220000">Sidi Bel Abbes</ville>
  </adresse>
  <remarque>
    Enseigne <clé>XML</clé> à l'ESI de Sidi Bel Abbes
  </remarque>
</carteDeVisite>
```

Prologue

commentaire

Élément racine

Élément enfant

Attribut

Le prologue

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes“ ?>
```

- Doit être positionné en toute première ligne du document XML.
- Les attributs "version", "encoding" et "standalone " doivent être places dans cet ordre ;
- N'est pas obligatoire et, dans ce cas, le parseur utilise un comportement par défaut (version 1.0, encoding UTF-8, standalone="yes“).
- Contient d'autres élément (que nous verrons plus tard)
  - Type de Document (DOCTYPE) pour les grammaires
  - Instructions de traitement (Stylesheet) pour les feuilles de styles

Le prologue

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
```

version : Spécifie la version de la norme XML utilisée (1.0 ou 1.1 qui sont très similaires). La version 1.1 est plus claire concernant certains nouveaux caractères unicode qui peuvent spécifier des retour à la ligne et autres caractères de contrôle.

encoding : Spécifie le jeu de codage de caractères utilisé.

- pour le français est le ISO-8859-1.
- Par défaut, l'attribut encoding a la valeur UTF-8 (Unicode).

standalone : Indique la dépendance du document par rapport à une déclaration de type de document.

- La valeur **yes**: le processeur de l'application n'attend aucune déclaration de type de document extérieure au document.
- La valeur **no**: le processeur attend une référence de déclaration de type de document.
- La valeur par défaut est **yes**.

Les nœuds éléments

- Les éléments définissent la structure du document.
- Un élément est délimité par:

<element> : balise ouvrante

</element> : balise fermante.

Exemple :

<adresse> 5, rue 1er Novembre 1954 </adresse>

Remarque:

Si une balise est vide (i.e. n'englobe pas de texte) alors elle pourra être simplifiée en une balise auto-fermante et s'écrira: <element/>

Exemple :

<adresse/>

Contenu d'élément

Du texte	<adresse> 5, rue 1er Novembre 1954 </adresse>
Des éléments	<adresse> <num> 5 </num> <rue>1er Novembre 1954 </rue> </adresse>
Contenu mixte	<adresse> <num> 5 </num> <rue>1er Novembre 1954 </rue> Sidi Bel Abbes </adresse>

Eléments : syntaxe générale

- Une balise doit être nommée :
  - Avec des lettres minuscules (c'est mieux), accentuées si l'on veut,
  - Éventuellement, des _ , des -, des ., des :, et des chiffres,
  - le nom doit toujours contenir au moins une lettre,
  - le premier caractère doit être alphabétique ou un tiret-souligné,
  - Le nom ne peut commencer par « XML » ,
- Aussi:
  - Les balises doivent être correctement imbriquées(pas de chevauchement).
    - **<P> bla <B> bla </B> </P>** est correct, alors que
    - **<P> bla <B> bla </P> </B>** ne l'est pas
  - Toute balise ouverte doit être fermée **<balise>…</balise>**
  - XML est sensible à la case (**Titre** ≠ **titre** ≠ **TITRE**)

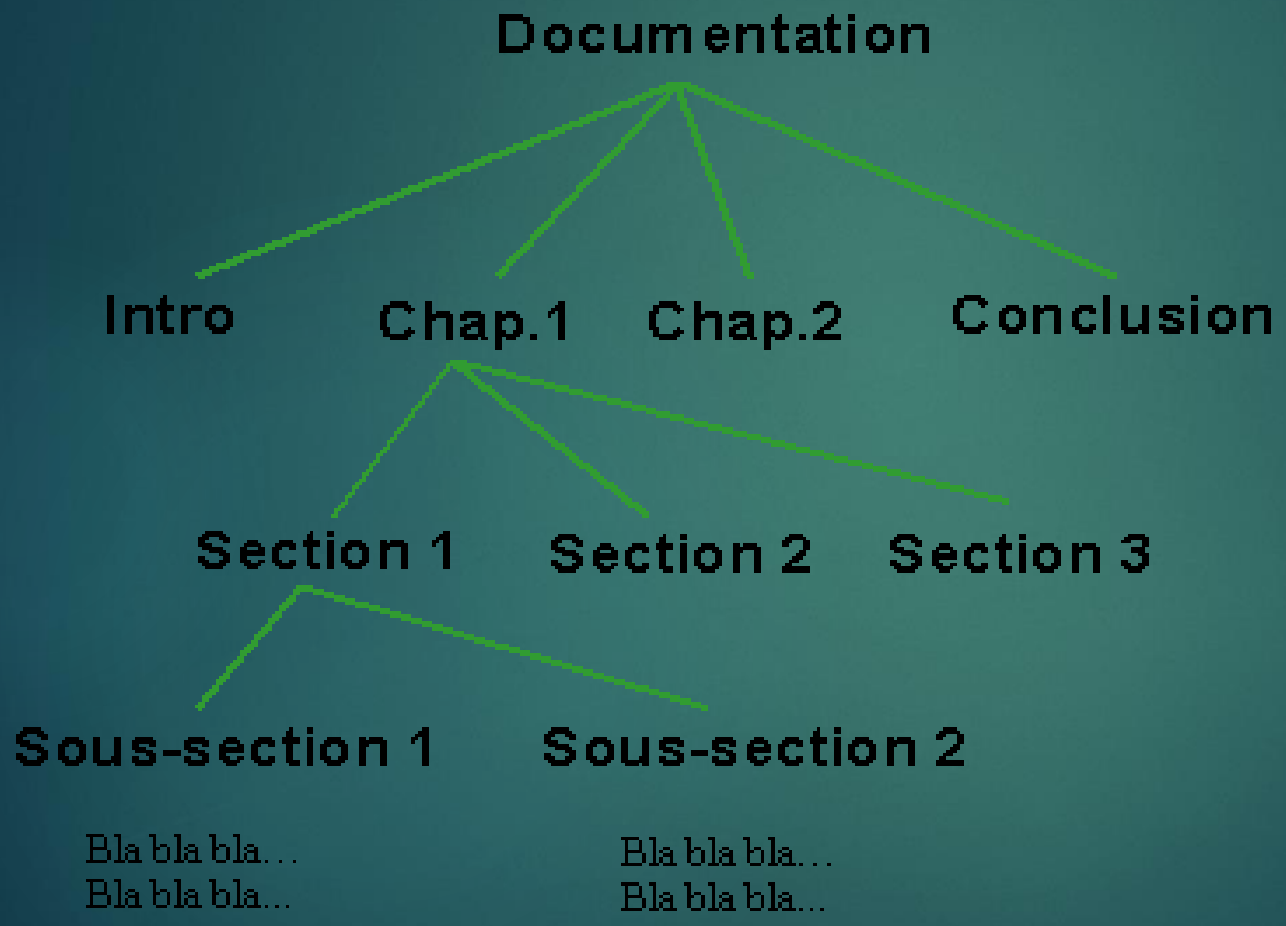
Syntaxe des noms d'élément (exemples)

1. nom
2. prénom
3. First name
4. Nom/prenom
5. 1234
6. à-la-ligne
7. xsl:value-of
8. 2010-catalogue
9. n123
10. décompte.client
11. first_name
12. xmlSpécication
13. _Ali
14. الإسم

Caractérisation d'un document XML

- Un document XML est dit **bien formé (Well-Formed)** s'il respecte les règles suivantes:
  - Le document doit contenir un élément racine (Root Element).
  - Tous les autres éléments doivent être contenus à l'intérieur des balises de l'élément racine et ils doivent être imbriqués correctement.
  - Le document doit respecter les règles de syntaxe XML.

Structure d'un document
Hiérarchie de composition des éléments




```
<Documentation>
    <intro>
    </intro>
    <Chap1>
        <Section1>
            <Sous-section1>
                Bla bla bla
            </Sous-section1>
            <Sous-section2>
                Bla bla bla
            </Sous-section2>
        </Section1>
        <Section2>
        </Section2>
        <Section3>
        </Section3>
    </Chap1>
    <Chap2>
    </Chap2>
    <Conclusion>
    </Conclusion>
</Documentation>
```

Structure: *Élément racine*

- Tous les documents XML doivent avoir une balise UNIQUE d'ouverture et de fermeture appelé l'élément racine (**root element**)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<livres>
<livre>
   <titre>HTML 4, XML et Java 2</titre>
   <auteurs>
      <nom_auteur>Eric Ladd   </nom_auteur>
      <nom_auteur>Jim O'Donnel</nom_auteur>
   </auteurs>
</livre>
<livre>
   <titre> XML cours et exercices </titre>
   <auteurs>
      <nom_auteur> Alexandre Brillant </nom_auteur>
   </auteurs>
</livre>
</livres>
```

- Tous les autres éléments sont contenus à l'intérieur des balises de l'élément racine; ce sont les sous-éléments **(child elements).**

Structure: *Élément racine*

- Sert à donner au programme qui analyse le document XML un point de référence.
- Exemple

Oui

```
<?xml version="1.0"?>
<root>
    <p>Hello World</p>
</root>
```

Non

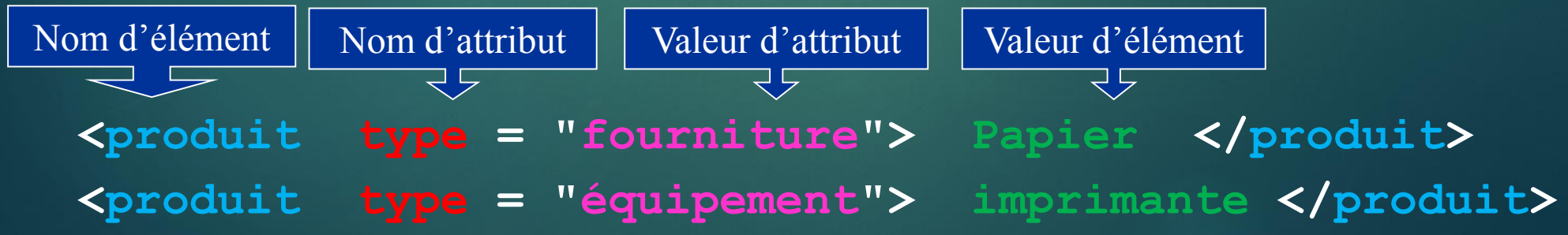
```
<?xml version="1.0"?>
<root>
    <p>Hello World</p>
</root>
<root>
    <p>Hello World2</p>
</root>
```

Les Attributs

- Les attributs sont toujours associés aux éléments. Ils viennent en quelque sorte les qualifier (Fournissent des informations supplémentaires sur l'élément),
- Les attributs sont codés au sein de la balise ouvrante de l'élément.
- L'ordre des attributs n'a pas d'importance.

Syntaxe : `nom_attribut="valeur"` oubien `nom_attribut='valeur'`

Exemple



Les Attributs

- ***Attention*** : Un attribut pour un élément donné ne peut avoir qu'une seule valeur

- *Exemple*

```
<produit prix="500" prix="200"> Papier </produit>
```

est interdit !

Élément VS Attribut

```
<biblio >
 <nblivre >2 </nblivre >
  <livre >
   <code >681.321 A9-CHA </code >
    <titre >Programmation HTML et JavaScript</titre >
    <auteur >
                <nom >Chaléat</nom >
                <prenom >Philippe</prenom >
    </auteur >
    <auteur >
                <nom >Charnay</nom >
                <prenom >Daniel</prenom >
     </auteur >
    <editeur >Eyrolles</editeur >
    <annee >2000</annee >
</livre >
  <livre >
    <code >681.321 XML-MIC </code >
    <titre >XML langage et applications</titre >
     <auteur >
                <nom >Michard</nom >
                <prenom >Alain</prenom >
     </auteur >
     <editeur >Eyrolles</editeur >
     <annee >2001</annee >
< livre >
</biblio >
```

```
<biblio nblivre="2" >
<livre
    Code="681.321 A9-CHA"
    titre="Programmation HTML et JS
    auteur_1="Chaléat Philippe"
    auteur_2="Charnay Daniek"
    editeur="Eyrolles"
    annee="2000"/>
<livre
    code="681.321 XML-MIC"
    titre="XML langage et applications"
    auteur="Michard Alain"
    editeur="Eyrolles"
    annee="2001"/>
</biblio>
```

Dans cet exemple la structure des auteurs en nom et prénom est perdue.

Dès qu'il y a plusieurs auteurs, il faut créer de nouveaux noms d'attributs pour les auteurs.

Élément VS Attribut

- Il n'existe pas de règle générale pour décider si telle information doit être mémorisée sous forme d'élément ou d'attribut.
- Le créateur du document ou du modèle est guidé par l'usage auquel est destiné le document.
- D'une manière générale les contenus d'éléments ont plutôt vocation à être des données, alors que les attributs sont plutôt destinés à mémoriser les caractéristiques des éléments, ou des informations pour les gérer comme des identifiants.

Évitons les attributs

- Ils ne peuvent pas contenir de valeurs multiples (les éléments oui).
- Ne sont pas facilement extensible (futur).
- Ne peuvent pas décrire les structures.
- Ils sont plus difficile à manipuler par les programmes.
- Peuvent être remplacés par des éléments.

```
<produit  type="papier" >
     <prix> 300  </code>
      ...
</produit>
```

```
<produit>
   <prix> 300  </prix>
   <type>  papier  </type>
   ...
</produit>
```

Règle sur le texte

- Les caractères **<** , **>** , **&** , **'** , et **"** ne peuvent pas être utilisés dans le texte (car utilisés dans le balisage) :
  - **<** à la place de **<**
  - **>** à la place de **>**
  - **&** à la place de **&**
  - **'** à la place de **'**
  - **"** à la place de **"**

<calcul>
if (a<b et b>c) ...
</calcul>

<calcul>
If (a<b et b>c)
</calcul>

Les commentaires

- Les commentaires XML se déclarent de la même manière qu'en HTML :
  - ils commencent par <!--
  - ils se terminent par -->
- Ils peuvent être placés n'importe où, du moment qu'ils sont à l'extérieur d'une autre balise.

Exemple de commentaires valides :

```
<!--commentaire correct -->
<texte><!-- commentaire correct   >
petit texte</texte>
```

Espaces de noms XML

Définition

- Un *espace de nom* a été introduit en XML afin de pouvoir mélanger plusieurs vocabulaires au sein d'un même document. Il permet de garantir l'unicité des noms d'éléments et d'attributs.
- Un espace de noms est déclaré par un pseudo attribut dont le nom prend la forme xmlns:prefix où prefix est un nom XML ne contenant pas le caractère ':'
- Un *espace de noms* est identifié par une URL(Uniform Ressource Locator).

Syntaxe

```
<prefix:element xmlns:prefix="URL">
```

Exemple

```
<cont:contact xmlns:cont="www.tutorialspoint.com/profile">
```

Espaces de noms XML

Document qui donne la position géographique d'un parc d'ordinateurs

```
<machines>
  <machine>
    <nom>unix1</nom>
    <adresse>
      <batiment>12</batiment>
      <bureau>R-09</bureau>
    </adresse>
  </machine>
….
</machines>
```

Document qui donne les adresses réseau d'un parc d'ordinateurs

```
<reseau>
  <equipement>
    <adresse>
      <IP>213.140.115.10</IP>
      <mac>01:00:5E:01:01:01</mac>
    </adresse>
  </equipement>
….
</reseau>
```

Espaces de noms XML

Ambigüité

```
<machines>
    <machine>
        <nom>unix1</nom>
        <adresse>
            <batiment>12</batiment>
            <bureau>R-09</bureau>
        </adresse>
        <adresse>
            <IP>213.140.115.10</IP>
            <mac>01:00:5E:01:01:01</mac>
        </adresse>
    </machine>
    ....
</machines>
```

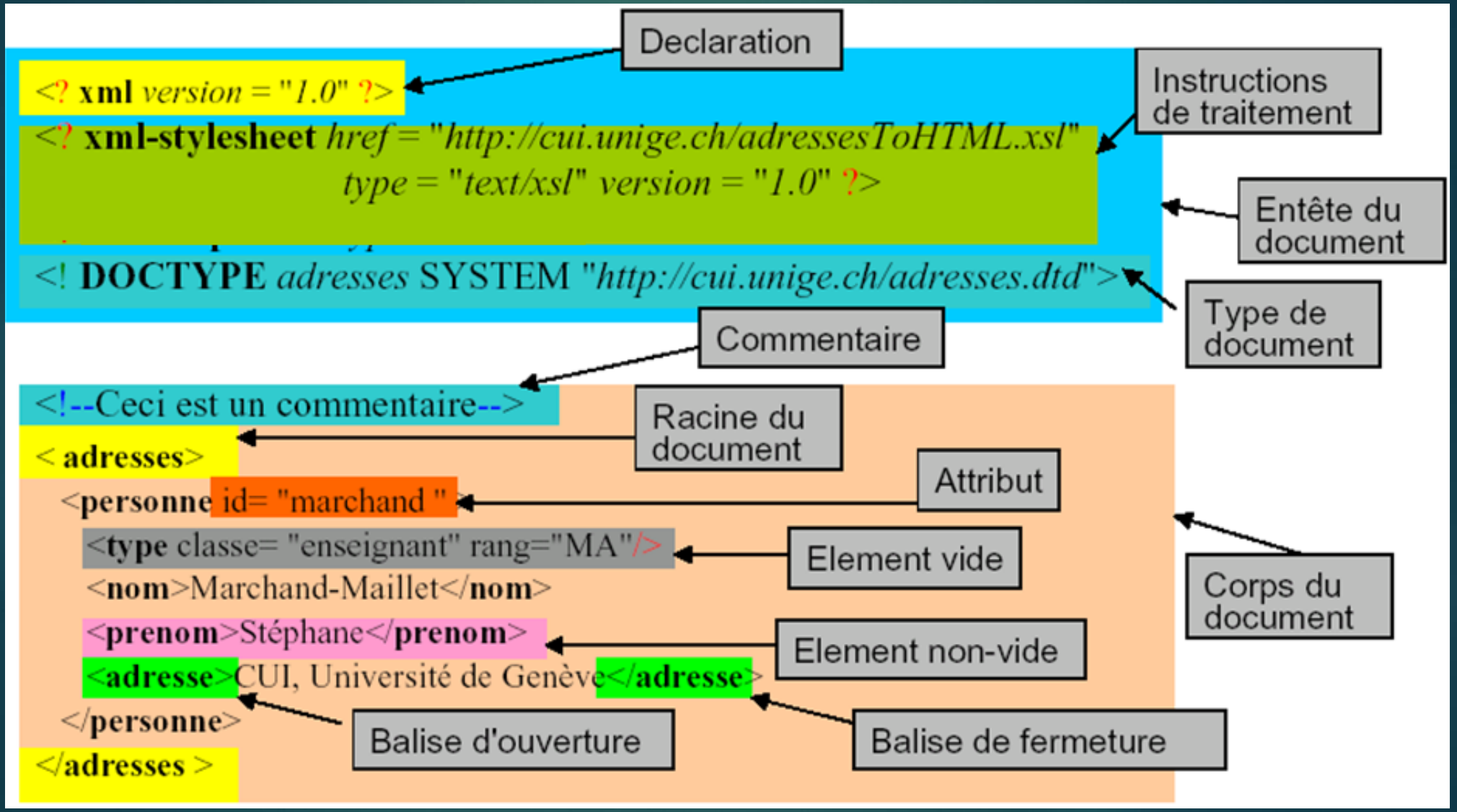
Espaces de noms XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<machines  xmlns:reseau="http://www.sba.dz/res" xmlns:local="http://www.sba.dz/loc">

    <machine>
        <nom>unix1</nom>
        <local:adresse>
            <local:batiment>12</local:batiment<batiment>
            <local:bureau>R-09</local:bureau>
        </local:adresse>
        <reseau:adresse>
            <reseau:IP>213.140.115.10</reseau:IP>
            <reseau:mac>01:00:5E:01:01:01</reseau:mac>
        </reseau:adresse>
    </machine>
      ....
</machines>
```

Exemple complet



XML : Les principaux atouts

- **Lisibilité** : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML
- **clarté**: séparation du fond et de la forme
- Une **structure arborescente** : permettant de modéliser la majorité des problèmes informatiques
- **Universalité** et **portabilité** : les différents jeux de caractères sont pris en compte; format texte (rien en binaire); il peut être facilement distribué par n'importe quels protocoles, comme HTTP
- **Intégrabilité** : un document XML est utilisable par toute application pourvue d'un « *parser* » (c'est-à-dire un logiciel permettant d'analyser un code XML)
- **Modularité**: un document XML doit pouvoir être réutilisable;
- **Extensibilité:** XML est un métalangage dont les bases peuvent être utilisées pour créer d'autres langages
- **Sécurité** : encryption, signature

XML : Les Inconvénients

- Format texte → XML est bavard comparé au format binaire: ce qui le rend très lourd à stocker et à transmettre.
- Récent → encore moyennement supporté par les applications.
  - Depuis 2014, tous les virements et prélèvements au sein de la zone euro se font avec la **norme SEPA** (Single Euro Payments Area) qui se base essentiellement sur XML
- Ça fait beaucoup de langages XML à connaître (XML, SCHEMA-XML, XSL, XSLT, XPATH, Xpointer, Xquery, etc.)

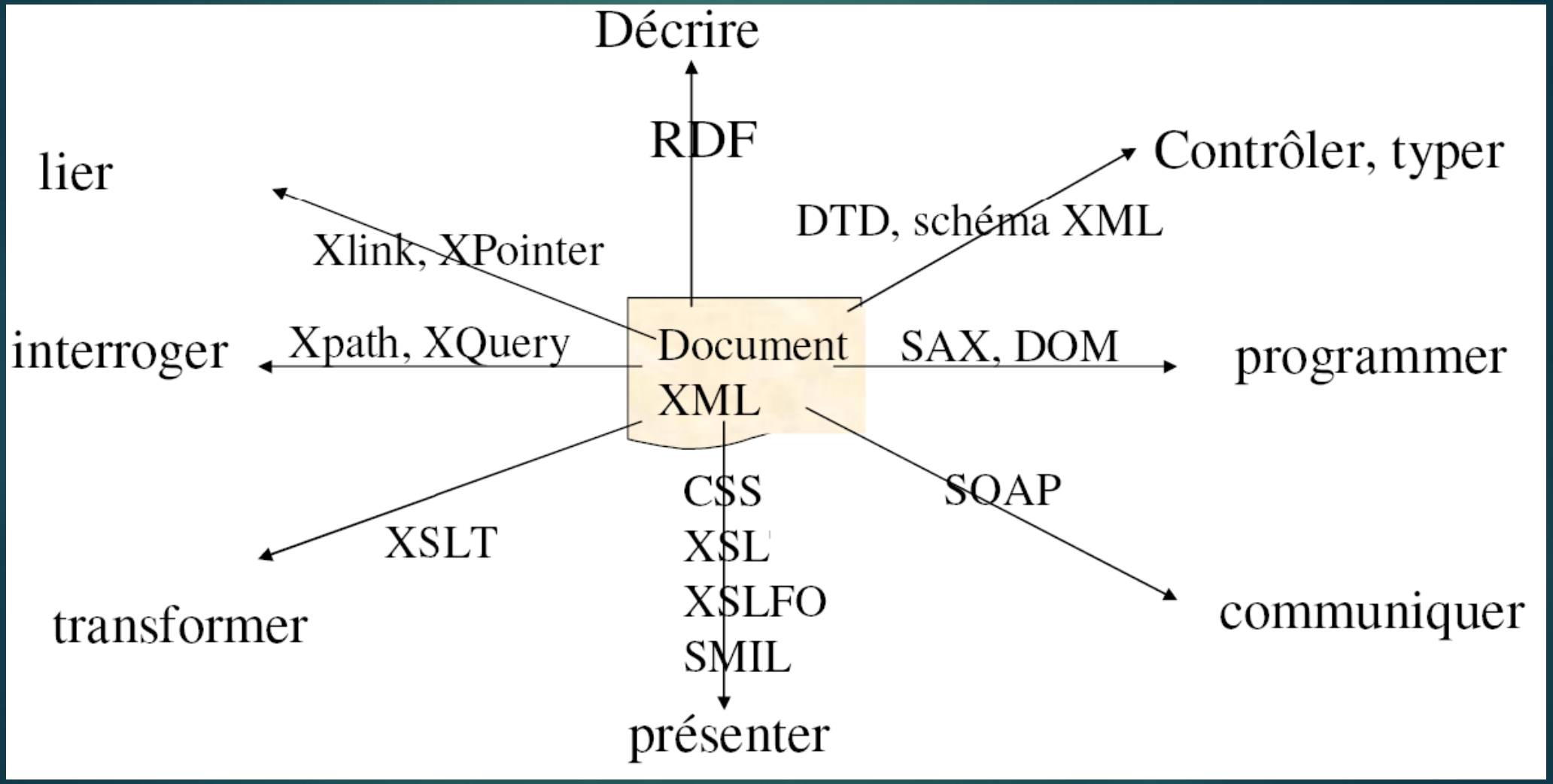
Applications XML

- Diffusion de contenu sur le web :
  - complément à HTML, à CSS …
- Échange entre applications distribuées
  - Services Web, …
- Archivage :
  - Base de données XML
- Gestion de la connaissance :
  - Ontologies

Editeur XML

- **Oxygen: C'est un soft développé en Java, disponible sur Windows et Unix et possédant une large gamme d'outils**
- BonFire Studio : très facile à utiliser, fonctionnalités de bases mais suffisantes pour mes besoins.
- Xml-Spy: C'est un peu plus qu'un éditeur xml, il est payant mais existe en version d'essai
- Cooktop: Il est entièrement gratuit et offre pas mal de possibilités.
- XMLwriter: Il est stable et facile à utiliser,
- XmlBuddy: sous Eclipse,
- **BaseX: est un gestionnaire de bases de données XML,**
- XMLCopy Editor : est un éditeur XML léger et rapide.

Les standards XML



Bibliographie

- XML cours et exercices / ELEXANDRE BRILLANT
- BASES DE DONNEES XQuery pour interroger des donnees XML / JACQUES LE MAITRE
- les cahiers du programmeur POSTGRE SQL / STEPHANE MARIEL
- INTRODUCTION AU XML / SIMON ST-LAURENT
- JAVA/XML / FLEURY RENAUD
- JAVA ET XSLT / BURKE ERIC M
- MODELISATION XML / ANTOINE LONJON
- ORACLE ET XML / MUENCH STEVE
- SQL POUR ORACLE/6eEDITION / SOUTOU CHRISTIAN
- XML / TITTEL ED

Bibliographie

- Kevin Williams, Michael Brundag XML et les bases de données, Eyrolles, 2001
- A. BRILLANT, XML cours et exercices, Eyrolles, 2010
- F. Knab, F. Lepoivre, F. Rivard, XML et Java, Eyrolles, 2000
- G. CHAGNON , F. NOLOT , XML : synthèse de cours et exercices corriges, PEARSON 2007
- Patrick Fabre, Xml, Ediscience, 2003
- J. Le Maitre, BASES DE DONNEES XQuery pour interroger des donnees XML, ellipses, 2013
- S. St-Laurent, Introduction Au XML, OSMAN EYROLLES MULTIMEDIA, 2000.
- Fleury Renaud, JAVA/XML, EYROLLES, 2005
- A. Lonjon, MODELISATION XML, EYROLLES, 2006
- Michard. XML langage et applications. Eyrolles. Paris. 2001.
- Marchal. XML by Example. Macmillan Couputer Publishing. 2000.
- M. Morrison. XML. CampusPress. 2005.
- F. Role. Modélisation et manipulation de documents XML. Lavoisier. 2005.
- M. Kay. XPath 2.0 Programmer's Reference. Wiley Publishing, Inc.. Indianapolis. 2004.
- Schémas XML V. Lamareille. XML Schema et XML Infoset. Cépaduès. 2006.
- J.-J. Thomasson. Schémas XML. Eyrolles. 2003.
- P. Drix. XSLT fondamental. Eyrolles. 2002.
- M. Kay. XSLT 2.0 Programmer's Reference. Wiley Publishing Inc.. 2004.
- M. Kay. XSLT 2.0 and XPath 2.0. Wiley Publishing, Inc.. Indianapolis. 2008.
- J-M Chauvet, Services Web avec SOAP, WSDL, UDDI, ebXML Eyrolles éditions, 2002
- Georges Gardarin. Des bases de données aux services Web, Dunod, Paris, 2002
- Hubert Kadima et Valérie Monfort, Les services Web: techniques, démarches et outils, Dunod, Paris, 2003

Questions ?