

# TECHNOLOGIES XML

Master IT 2013/2014

Prof. M.D. RAHMANI

# PLAN

- Les bases de XML

Syntaxe, éléments, attributs, arbres, document bien formé,  
Espace de noms

- Les Grammaires XML

Document XML valide (DTD , XML Schéma)

- XML: recherches et transformation

XPath, XSLT

- Programmation XML

DOM, SAX

# BIBLIOGRAPHIE

- J.C.Bernadac, F. Knab, *Construire une application XML*, Ed. Eyrolles (1999).
- A. Michard, *XML, Langage et Applications*, Ed. Eyrolles (1999).
- S. Lecompte, T. Boulager,, *XML par la pratique (2<sup>ème</sup> edition)*, Ed. ENI

# WEBOGRAPHIE

- **Tous les liens sur XML et SGML:**

<http://www.oasis-open.org/cover>

- **Les recommandations W3C :**

- la page XML <http://www.w3.org/XML>
- la recommandation XML <http://www.w3.org/TR/REC-xml>

- **Cours XML :**

- Olivier carton

<http://www.liafa.jussieu.fr/~carton/Enseignement/XML/Cours/support.pdf>

- Cours et tutoriels: <http://xml.developpez.com/cours>

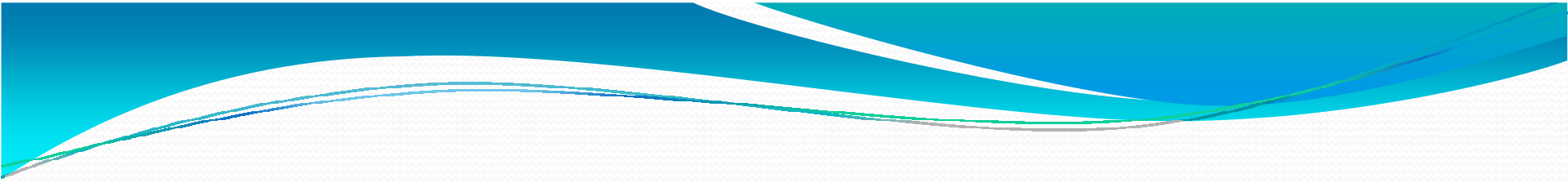
- Cours XML université Pierre et marie curie: <http://www.gchagnon.fr/cours/xml>

- Université Aix-Marseille

[http://www.dil.univ-mrs.fr/~tichit/java/CoursXML\\_1\\_XML.pdf](http://www.dil.univ-mrs.fr/~tichit/java/CoursXML_1_XML.pdf)

- Cours CNAM:

<http://www.infres.enst.fr/~talel/cours/bda/enst-xml-xpath-xslt-t3-02-03.pdf>



# Définitions

*XML est un méta-langage qui permet de créer des langages de balisage adaptés aux types d'informations à décrire*

*XML est un langage de description de document.*

# Historique

- 1986: **SGML** (*Standard Generalized e la structure d'un document*).  
Il propose à la fois une approche *Markup Language*) fut adopté par l'ISO pour normaliser la description d générale et ouverte, mais assez complexe à mettre en œuvre.  
( 20 % seulement de la fonctionnalité de SGML était utilisée).
- 1990 marque la naissance de **HTML** (*Hypertext Markup Language*)
- Le nouveau langage rendit nécessaire la création d'un nouveau protocole destiné au transfert des données HTML, c'est la naissance du **HTTP** (*Hypertext Transfert Protocol* ).
- Les serveurs http mettent les pages HTML à la disposition de l'utilisateur, les documents sont lus à l'aide des **navigateurs**.

# Historique

- 1993 apparut **MOSAIC**, le premier navigateur WEB avec interface graphique.
- A l'origine était sous UNIX, puis sous Windows, son concepteur Marc Anderesen est le fondateur de la société **NETSCAPE**.
- 1996: Le consortium W3C (*Word Wide Web Consortium*) et les grandes entreprises des logiciels se sont mis d'accord pour définir un langage de balisage normalisé.
- 1998: La 1<sup>ère</sup> version 1.0 des spécifications de **XML** (*eXtensible Markup Language* ou langage de balisage extensible) a fait l'objet d'une recommandation XML 1.0.

# Caractéristiques de XML

- Tout comme son parent SGML, XML est un **méta-langage**.  
XML permet la définition d'un langage pour décrire la structure de l'information associée à un type de document donné.
- Le mot **markup** rend compte du fait que XML permet d'identifier les différents éléments d'un document au moyen de balises
- HTML n'est qu'une application de SGML avec une syntaxe figée et non extensible.
- XML est **extensible**: on peut créer autant de balises que l'on souhaite.  
Les balises HTML servent à formater les informations.  
Les balises XML ne servent qu'à structurer les documents.

Remarque: *Un document XML ne possède aucune information sur la manière dont il doit être affiché sur un écran d'ordinateur.*

# Editeurs XML

- **Notepad++:** <http://notepad-plus-plus.org/fr/>
- **jEdit:** <http://www.jedit.org/>
- **XMLWriter:** <http://xmlwriter.net/>
- **Eclipse:** <https://www.eclipse.org/>
- **Editix XML Editor:** <http://www.editix.com/>
- **Oxygen XML Editor:** <http://www.oxygenxml.com/>

# Exemple d'un document XML

## demo.xml

- 1- `<?xml version="1.0"?>`
- 2- `<demoXML>`
- 3- `<message>Voici du XML</message>`
- 4- `</demoXML>`

## Explication:

- 1- prologue: déclaration du type de document.
- 2- la balise ouvrante de l'élément racine du document
- 3- un sous élément avec un contenu
- 4- la balise fermante de l'élément racine du document

# Exemple: demo1.xml

## 1. PROLOGUE

**<?xml version='1.0' encoding='ISO-8859-1' ?>**

déclaration, encodage ISO-Latin

**<!-- fichier demo.xml -->**

commentaire

la DTD, s'il y a, est dans le prologue

## 2. ARBRE d'ELEMENTS

**<demoXML>**

élément racine du document

**<message>Voici du XML</message>**

élément = balise d'ouverture + contenu + balise de fermeture

balise d'ouverture peut contenir des attributs

**</demoXML>**

# LE DOCUMENT XML

## I- Le prologue:

1- Une déclaration qui annonce qu'il s'agit d'un document XML.

**<?xml version="1.0" [encoding="chaîne"] [standalone="yes|no"]?>**

**<?xml ?>** est une déclaration

- l'attribut **encoding** précise l'encodage des caractères du document : "UTF-8", "UTF-16", "US-ASCII" l'ascii sans caractère accentué, "ISO-8859-1" français.

- **standalone** précise si le document est seul (sans DTD) ou non (auquel cas une instruction **<!DOCTYPE ...** est nécessaire)

# LE DOCUMENT XML

2- instructions de traitement :

`<?application argument1 arg2 ... argN?>`,

qui ne font pas parties du document au sens XML.

Ce sont des instructions au processeur XML.

**Exemple:**

`<?stylesheet TYPE="text/css" HREF="feuille.css"?>`

Il s'agit dans ce cas d'associer une feuille de style CSS au document XML

3- commentaire

`<!-- commentaire -->`

# LE DOCUMENT XML

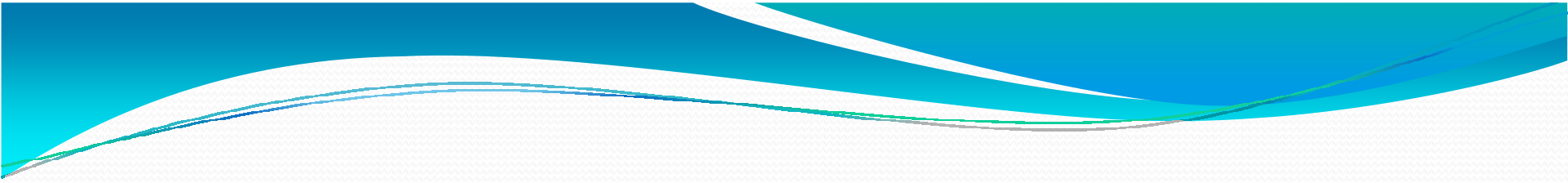
## II- Arbre d'éléments:

- Le **contenu** est structuré en **éléments** qualifiés par des **attributs** avec des **valeurs**
- Un **élément** est représenté par une paire de balises et son contenu
- Les balises décrivent la structure du texte, pas sa présentation
- Les balises ouvrantes peuvent porter zéro, un ou plusieurs **attributs** qui donnent des précisions sur les éléments et leur contenu
- Un **attribut** a un **nom** et une **valeur**
- **Eléments** et **attributs** ne sont pas prédéfinis, mais choisis en fonction du type de document à représenter
- Le balisage est indépendant du système où il est créé et des traitements à effectuer

# LE DOCUMENT XML

## Arbre d'éléments:

- un seul élément racine du document,
- les balises doivent être correctement emboîtées,
- les noms des éléments et attributs suivent la spécification des "namespaces", espaces de noms.
- **<texte/></b> élément vide est équivalent à **<texte></texte></b>****
- **<payement devise='dirham' mode="CB"></b> balise d'ouverture avec attribut(s) : couple **nom='valeur'</b> ,****
- les valeurs d'attribut sont entre " ou entre '



# LE DOCUMENT XML

## Attributs prédéfinis :

- **xml:lang="fr"** indication de la langue
- **xml:space="default|preserve"** indique au processeur XML ce qu'il faut faire des "blancs" (espace, tabulation, nouvelle ligne)  
les supprimer ou les sauvegarder.

# LE DOCUMENT XML

## Les entités:

Une entité est une chaîne de caractères commençant par **&** et se terminant par **;**

Elle s'apparente à une macro

➤ Référence à une entité prédéfinie :

**&lt;** référence l'entité prédéfinie **<**

**&gt;** pour **>**

**&amp;** pour **&**

**&quot;** pour **"**

**&apos;** pour **'**

**&#valeur;** code un caractère, par exemple **&#x20;** pour l'**espace**

# LE DOCUMENT XML

## La section CDATA:

Une section CDATA (*Character Data*) contient une chaîne de caractères qui n'est pas analysée par le processeur XML

Exemple: demo2.xml

**<texte>**

le code que vous devez saisir est : `<![CDATA[a<b et b>c]]>`  
pour entrer

**</texte>**

**a<b et b>c** est une section littérale (constante), donc elle n'est pas "interprétée".

# Définition d'attributs

Attribut or pas ?

La construction:

**<replique>**

**<personnage attitude='assis souriant' geste='un dollar entre les mains'>***Jordan Belfort***</personnage>**  
**<texte ton='long' >***Je m'appelle Jordan Belfort. L'année de mes 26 ans, à la tête de ma propre firme de courtage, je me suis fait 49 millions de dollars, ce qui m'énerve vraiment car ça faisait à peine un million par semaine.***</texte>**  
**</replique>**

pourrait être remplacée par :

**<replique>**

**<personnage>***Jordan Belfort***</personnage>**  
**<attitude>***assis souriant***</attitude>**  
**<geste>***un dollar entre les mains***</geste>**  
**<ton>***long***</ton>**  
**<texte>***Je m'appelle Jordan Belfort. L'année de mes 26 ans, à la tête de ma propre firme de courtage, je me suis fait 49 millions de dollars, ce qui m'énerve vraiment car ça faisait à peine un million par semaine.***</texte>**  
**</replique>**

*Jordan Belfort (Leonardo DiCaprio) dans Le Loup de Wall Street de Martin Scorsese, 2013*

# Attributs ou éléments

- Il faut plutôt utiliser un élément
  - lorsque l'ordre est important (l'ordre des attributs est au hasard)
  - lorsqu'on veut réutiliser un élément plusieurs fois (avec le même parent)
  - lorsqu'on veut (dans le futur) avoir des descendants / une structure interne
  - pour représenter un type de données (objet) plutôt que son usage, autrement dit: ***une "chose" est un élément et ses propriétés sont des "attributs"***
- Il faut plutôt utiliser un attribut
  - lorsqu'on désire faire référence à un autre élément
  - pour indiquer l'usage/type/etc. d'un élément  
<adresse usage="prof"> ... </adresse>
  - lorsqu'on veut imposer des valeurs par défaut dans la DTD

# Documents bien formés

Un document XML est bien formé (*well-formed*) s'il respecte les règles de la syntaxe XML :

- Un document possède une racine et une seule
- Tout élément a une balise ouvrante et une fermante : `<n>...</n>`  
Eléments vides : `<b></b>` ou `<b/>`
- Les paires de balises ouvrantes/fermantes sont imbriquées  
`<a>...<b>...</b>...</a>`  
`<a>...<b>...</a>...</b>` × pas de chevauchement
- Chaque attribut doit avoir une valeur
- Un attribut ne peut apparaître qu'une fois pour un même élément
- Un processeur XML doit s'arrêter lorsqu'il rencontre une erreur

# Arbres XML

- Un document bien formé peut être considéré et traité comme un arbre

```
<a>
  <a>
    <b>bla</b>
    <c>
      <a><b/>...</a>
    </c>
  </a>
</c/>
</a>
```

- L'imbrication et l'ordre des éléments reflètent la structure arborescente et ordonnée
- Les éléments terminaux sont des éléments vides ou des chaînes de caractères
- Particularité : les attributs d'un même élément n'ont pas d'ordre

# Formatage d'un document XML

## CSS:

- CSS1 et CSS2 (*Cascading Style Sheets*) sont des feuilles de style dédiées à la présentation visuelle de documents XML.

## XSL:

Le langage XSL (*Extensible Stylesheet Language*), se compose de deux sous langages:

- 1- **XSLT**: un langage permettant d'effectuer des transformations sur un document XML
- 2- **XSLFO**: un langage permettant le formatage de documents XML.

## API:

Une autre façon de formater un document consiste à modifier son contenu par un programme (Java, C++, PERL,...), pour aboutir à un document HTML ou un format de sortie recherchée.

Les deux principales catégories d'API sont (SAX et DOM)

# Espaces de noms XML

## Définition:

Un espace de noms permet de garantir l'unicité des noms d'éléments et d'attributs.

On attache une information unique à un nom de balise.

Cette information est une URL(*Uniform Resource Locator*).

## Déclaration:

**xmlns:reseau="http://www.fsr.ac.ma/reseau"**

# Espaces de noms XML

- Soit un 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>

- Un document qui donne les adresses réseau d'un parc d'ordinateurs:

<reseau>

<equipement>

<adresse>

<IP>212.140.115.10</IP>

<nom>unix1.fsr.ac.ma</nom>

</adresse>

</equipement>

....

</reseau>

Remarque: la même balise **adresse** n'a pas le même sens dans les deux cas.

# Espaces de noms XML

Si on fusionne les 2 documents, la signification de la balise **adresse** devient ambiguë:

```
<machines>
  <machine>
    <nom>unix1</nom>
    <adresse>
      <batiment>12</batiment>
      <bureau>R-09</bureau>
    </adresse>
    <adresse>
      <IP>212.140.115.10</IP>
      <nom>unix1.fsr.ac.ma</nom>
    </adresse>
  </machine>
  ....
</machines>
```

# Espaces de noms XML

La solution est d'associer un **espace de noms** et un **préfixe** qui qualifie les noms des éléments et des attributs de cet espace.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<machines>
  <machine>
    <nom>unix1</nom>
    <adresse>
      <batiment>12</batiment>
      <bureau>R-09</bureau>
    </adresse>
    <reseau:adresse
      xmlns:reseau="http://www.fsr.ac.ma/reseau" >
      <reseau:IP>212.140.115.10</reseau:IP>
      <reseau:nom>unix1.fsr.ac.ma</reseau:nom>
    </reseau:adresse>
  </machine>
  ....
</machines>
```

# Espaces de noms XML

Plusieurs espaces de noms peuvent être utilisés.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<local:machines
  xmlns:reseau="http://www.fsr.ac.ma/reseau"
  xmlns:local="http://www.fsr.ac.ma/localisation">
  <local:machine>
    <local:nom>unix1</local:nom>
    <local:adresse>
      <local:batiment>12</local:batiment>
      <local:bureau>R-09</local:bureau>
    </local:adresse>
    <reseau:adresse>
      <reseau:IP>212.140.115.10</reseau:IP>
      <reseau:nom>unix1.fsr.ac.ma</reseau:nom>
    </reseau:adresse>
  </local:machine>
  ....
</local:machines>
```

# Espaces de noms XML

**Espace de noms et attribut:** Les attributs peuvent également bénéficier des espaces de noms

Exemple:

Soit:

```
<produit monnaie="Dirhams">  
  <prix>140</prix>  
</produit>
```

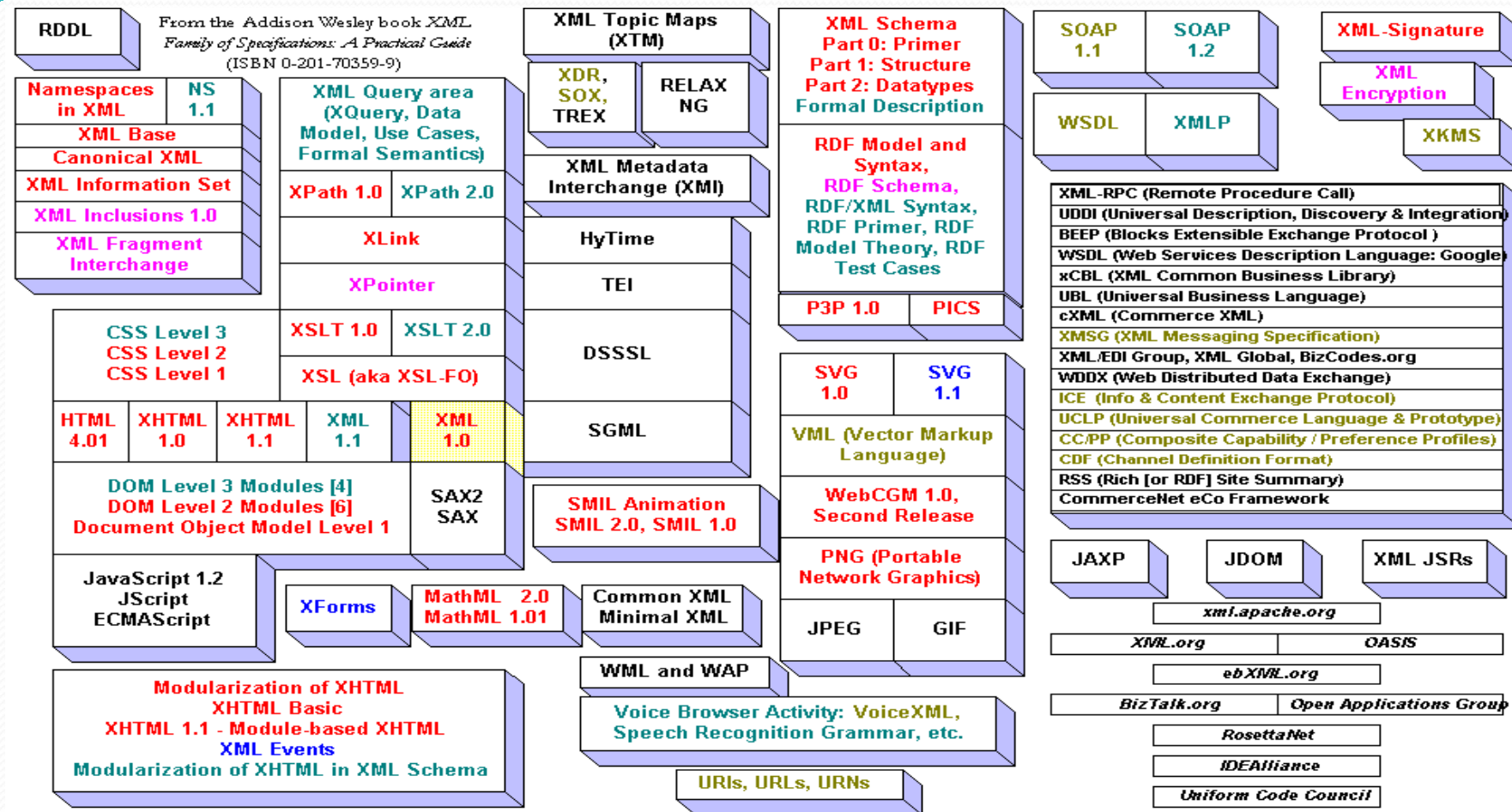
Devient:

```
<com:produit xmlns:com='http://www.societe.com/commercial'  
  com:monnaie = 'Dirhams'>  
  <com:prix> 150</com:prix>  
</com:produit>
```

# Dialectes, extensions et applications

- **XLink** et **XPointer**: Liens entre documents
- **XPath**: Sous-ensemble d'un document
- **XQuery**: Requêtes de base de données
- **XSL**: Transformation de documents
- **Schémas XML**: Modèles de documents
- **RSS** : (Really Simple Syndication) Flux de données
- **XUL** : (XML-based User interface Language) Langage de description d'interfaces graphiques
- **SVG** : (Scalable Vector Graphics) Dessin vectoriel
- **SMIL**: (Synchronized Multimedia Integration Language) Multimédia
- **MusicML**: partition musicale
- **CML**: description des molécules chimiques en 3D
- **MathML**: l'équivalent du **TEX** (formules mathématiques)

# La galaxie XML



## The XML Family of Specifications: The Big Picture

Last Updated: April 16, 2002



Copyright (c) 2002 Kenneth B. Sall. All Rights Reserved.