XML : eXtensible Markup Language

Plan

- XML un modèle de données
- XML et DTD
- XML et Xschema
- XML, Xpath et XSL

XML un modèle de données

Les BDs et le web

Exemple de formats existants

- Texte enrichi avec du formatage

*« Il y a **28** variétés de pommes en France, **76** aux USA et **3** en Chine. »*

RTF →

```
{{\rtf1\ansi {\i\f0
Il y a \b 28\b0
vari\'e9t\'e9s de
pommes en
france\par }
```

- Page Web

HTML →

```
<html>
  <body>
    <h1>
      Le Monde
    </h1>
  </body>
</html>
```

Publier des données dans le web

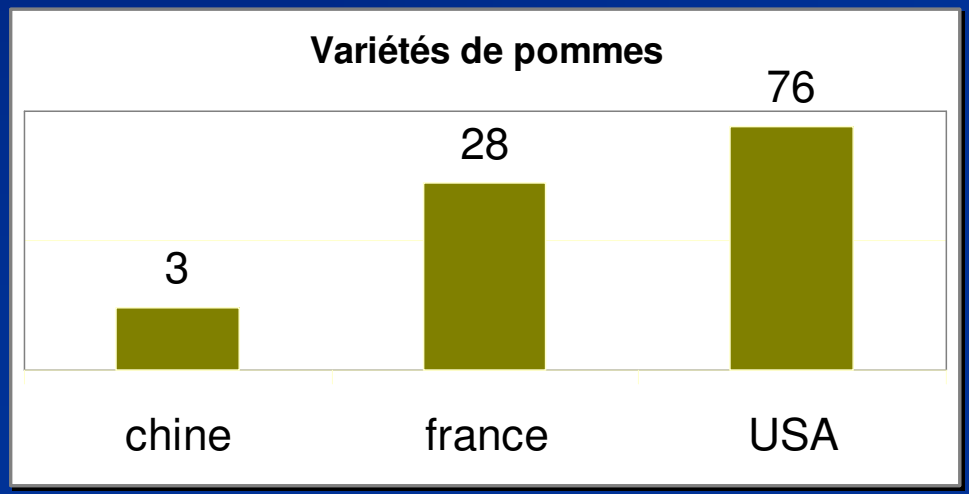
- texte formaté

*« Il y a **28** variétés de pommes en France, **76** aux USA et **3** en Chine. »*

- tableau

pays	variété
France	28
USA	76
Chine	3

- graphe



Mémoriser des données

- Données structurées
  - *ex* : feuille de calcul, transaction financière, dessin technique
- Stockage dans un fichier
  - format texte: lisible
  - format réutilisable
    - par plusieurs logiciels,
    - indépendant du logiciel initial
  - format extensible
    - international (ex: caractères chinois)
    - indépendant de la plate-forme

Formater les données

*« Il y a **28** variétés de pommes en France, **76** aux USA et **3** en Chine. »*

```
Il y a 28 variétés de
pommes en France, 76 aux
 USA et 3 en Chine.
```

gras

gras

phrase en italique

pays	variétés
france	28
USA	76
chine	3

```
pays variétés
france 28
USA 76
chine 3
```

en-tête jaune

tableau: 2 col. et 3 lignes

La nouveauté d'XML

- Evolution des langages de description de documents

XML ressemble à HTML

- Balisage ("marquage")
  - balise ouvrante, balise fermante: délimitent un élément
    - contenu d'un élément
  - HTML: balise de formatage
    - titre, sous-titre, paragraphe, tableau, liste à puces ...
  - XML: balise de structuration
    - Organise les éléments composant le document
- attributs
  - forme: <b **nom = ‘‘valeur’’** >
  - *ex:* prix= ‘‘10’’

Formatage en HTML

*« Il y a **128** fournisseurs d'accès en France, **7600** aux USA et **3** en Chine. »*

 Il y a 28 variétés de pommes en France, 76 aux USA et 3 en Chine.

Légende des balises:

 : bold (gras)

 : italique

Formatage en HTML et XML

en-tête

```
pays pommes
france 28
USA 76
chine 3
```

tableau: 2 col. et 3 lignes

HTML

```
<table border = "1" >
  <th> <td>Pays</td> <td> Pommes </td> </th>

  <tr>  <td> France </td>  <td> 28 </td>   </tr>
  <tr>  <td> USA    </td>  <td> 76 </td>   </tr>
  <tr>  <td> chine  </td>   <td> 3  </td>   </tr>
</table>
```

XML

```
<enquête>
  <vp nombre ="28">    <pays>france</pays>   </vp>
  <vp nombre ="76">  <pays>USA</pays>     </vp>
  <vp nombre="3">       <pays>chine</pays>    </vp>
</enquête>
```

Structure avec éléments et attributs

Plusieurs structures sémantiquement équivalentes

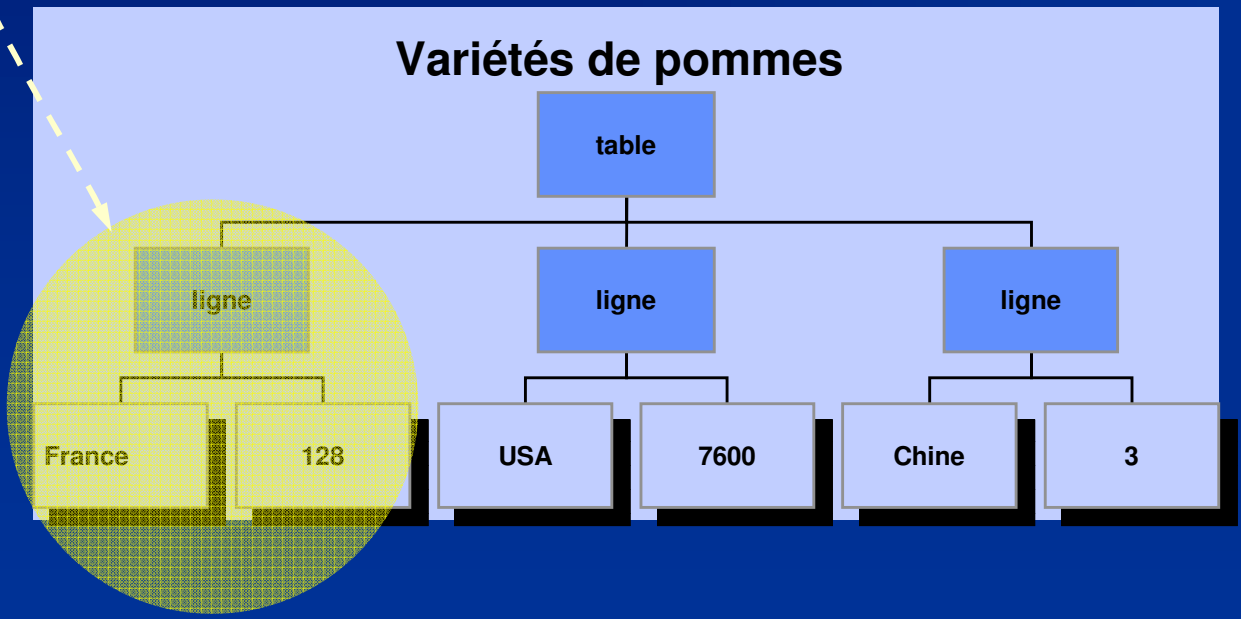
```
<enquête>
  <vp nombre =“28”>   <pays>France</pays>   </vp>
  <vp nombre =“76”>   <pays>USA</pays>      </vp>
  <vp nombre=“3”>     <pays>Chine</pays>    </vp>
</enquête>
```

```
<enquête>
   <vp nombre =“28”  pays="France" > </vp>
   <vp nombre =“76”  pays="USA" >  </vp>
   <vp nombre=“3”    pays ="Chine" > </vp>
</enquête>
```

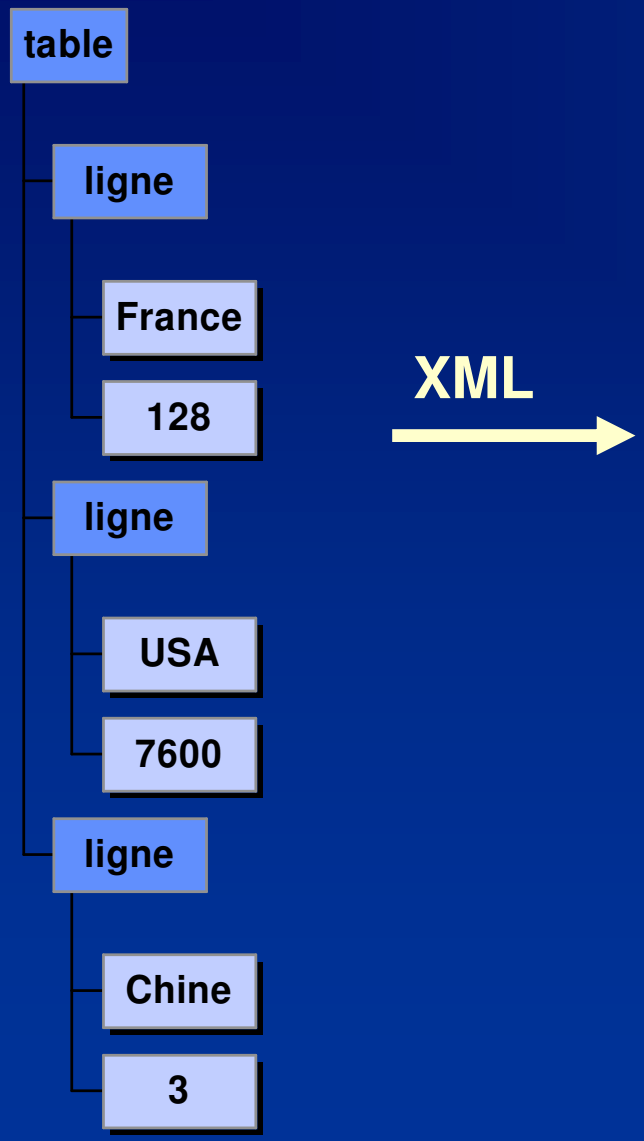
```
<enquête>
 <vp > 28 </vp>   <pays>france</pays>
 <vp > 76 </vp>   <pays>USA</pays>
 <vp > 3 </vp>    <pays>Chine</pays>
</enquête>
```

Modèle relationnel vers XML (1)

pays	pommes
france	28
USA	76
chine	3



Modèle relationnel vers XML (2)



XML

```
<table>
    <ligne>
        <pays>France</pays>
        <vp>28</vp>
    </ligne>

    <ligne>
        <pays>USA</pays>
        <vp>76</vp>
    </ligne>

    <ligne>
        <pays>Chine</pays>
        <vp>3</vp>
    </ligne>
</table>
```

XML : « successeur » de HTML

- HTML HyperText Markup Language.
  - Un ensemble prédéfini et limité de balises surtout de présentation, défini par une norme (HTML 2.0, 3.2, 4.0).
- Sémantiques des balises :
  - `h1`,..,`h6`, `title`, `address`, ... donnent des indications structurelles
  - `center`,`hr`,`b`,`i`,`big`,`small`,... ne servent qu'à décrire une mise en page
- Tim Berners-Lee (le créateur de HTML) a lui-même encouragé pour un successeur. Pourquoi?

Problèmes liés à HTML

- L'affichage d'un document est fortement dépendant de l'interprétation qu'en fait le navigateur
- Il est nécessaire de disposer de plusieurs versions du document en fonction du média de rendu
- L'indexation de documents ne peut se faire que sur la partie textuelle
- ***Document et pas donnée***

Différence entre HTML et XML

- XML est plus strict qu'HTML
  - Règles d'écriture strictes
    - syntaxe non ambiguë
    - pas d'erreur d'interprétation
  - Document avec des contraintes sémantiques
    - ex : un livre a un titre et un seul
    - vérification de la conformité du document
- XML est extensible
  - définition de nouvelles balises
  - inconvénient mineur: XML est verbeux
    - taille > fichier binaire
    - réduire la taille par compression (ex: zip)
    - et réduire le transfert: communication haut débit

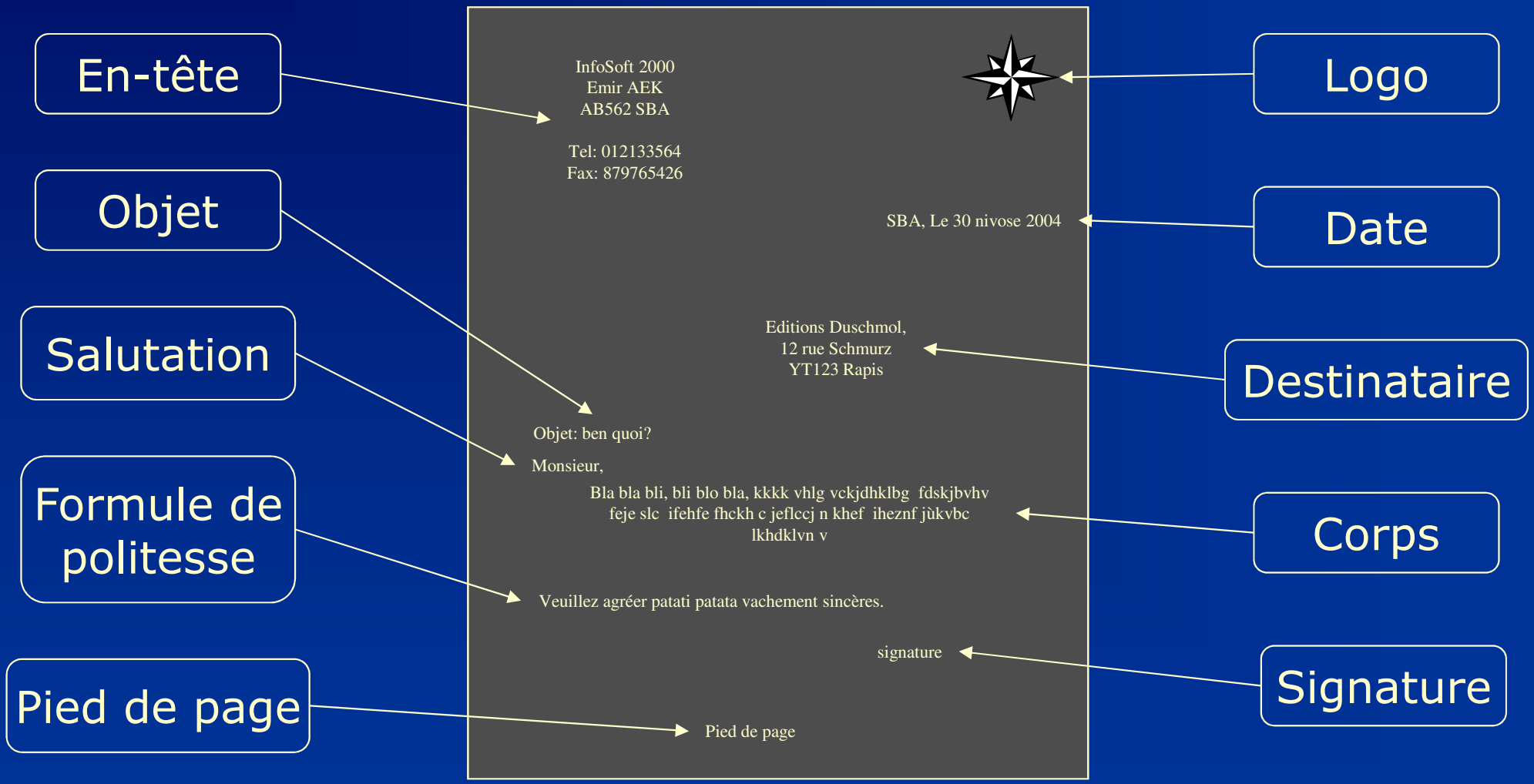
SGML et le balisage structurel

- Il fallait passer d'un de balisage de *présentation à un BALISAGE STRUCTUREL*
- XML comme SGML dont il est un descendant utilisent un balisage structurel
- SGML : Standardized Generalized Markup Language
  - Très utilisé en documentation technique
  - Airbus: la doc doit être précise et non ambiguë
- Ce sont des *métalangages* de *description* et *d'échange* de *documents structurés*
  - Métalangage: possibilité de définir des « dialectes » dans des domaines particuliers

XML contre SGML

- SGML norme ISO 8879:1986
- Très utilisé dans l'industrie pour de grandes documentations techniques.
- Trop complexe pour une utilisation « grand public » ou dans des domaines moins exigeants sur la précision
- SGML: trop de trucs compliqués et inutiles
- XML utilise 10% de SGML pour représenter efficacement la plupart des besoins des applications

Exemple de document



Représentation XML

```
<lettre>
<entete>
  <logo loc="logo-graph.vml"/>
  <adresse>
    &abrev-adresse;
  </adresse>
</entete>
<destinataire>
  <nom> Mr Badsi</nom>
  <adresse>
    <rue>
      Emir AEK
    </rue>
    <ville>
      SBA
    </ville>
  </adresse>
</destinataire>
<objet> bla bla </objet>
…
```

```
…
<date>
  30 Novembre 2015
</date>

<salutation>
  Monsieur,
</salutation>

<corps>
  <para>
    Ici le premier paragraphe
  </para>
  <para>
    et là le deuxième
  </para>
</corps>

</lettre>
```

Points importants

- La représentation de cette lettre en XML ne comporte aucune indication sur sa mise en page
  - Les aspects graphiques ou typographiques sont absentes du source XML
  - Ces aspects seront définis par l'intermédiaire d'une *feuille de style*
- Une feuille de style est un *ensemble de règles* pour spécifier la *réalisation concrète* d'un document sur un *média* particulier

XML est libre de droits

- XML est standard
  - W3C : consortium international
- Indépendance / produit
- Ressources:
  - logiciel libre
  - tutoriel libre
- Communauté de programmeurs
  - listes de discussion
  - échange de connaissances

XML : une famille de technologies

- Spécification XML1.0 du consortium W3
- Application d'XML
  - pour spécifier des langages
    - manipulation de données (XSL), navigation (XLL)
  - pour spécifier des formats d'échange
    - protocole : Simple Object Access Protocol
    - type de données
      - Wireless Markup Language, Scalable Vector Graphics
  - lien entre XML et les langages de programmation
    - Document Object Model pour java, C++, PHP, ...

Principe général

- Pour une application particulière
- On se définit une syntaxe: un dialecte XML
- On définit la sémantique de ce dialecte

- Pour me demander un rendez-vous, il faut m'envoyer le document xml-rdv du type suivant

  <rdv><d><n>$x</n> $y </d>
  <h232>$z</h232> $p </rdv>

  Où $x est votre nom, $y votre prénom, $z la date et l'heure du rdv au format ISO… et $p optionnel, un lieu de rdv.

Exemples de dialectes XML

- XHTML
- MathML
- SVG
- XSL
- SOAP
- WSDL
- XML Schema

XHTML = HTML avec un syntaxe XML

- Reformulation de HTML en tant qu'application XML
  - En gros: on ferme ce qu’on a ouvert…
- Intérêt
  - Syntaxe plus rigoureuse
  - Importation de fragments de documents d'autres domaines nominaux
  - Possibilité d'utiliser les applications XML standard

MathML : les maths en XML

- Permettre l'échange et le traitement d'expressions mathématiques sur le Web
- Insertion aisée d'expressions mathématiques dans des documents HTML ou XML
- Communication d'expressions entre applications au plan *sémantique*

SVG : le graphique 2D en XML

- Langage de description de graphiques 2D
- Graphiques vectoriels
- Interactifs et dynamiques
  - Animations déclaratives
  - Programmation ECMAScript
- Recommandation du 04/09/2001

SMIL

- Vidéo
- Synchronisation de l’image et du son
- Synchronisation entre plusieurs fenêtres

SOAP : calcul distribué en s'échangeant du XML

- Simple Object Access Protocol
- SOAP 1.1 : soumission au W3C du 08/05/2000
- Protocole d'échange de données entre applications distantes
- Adapté pour être utilisé au-dessus du protocole HTTP (méthode POST)
- Structure d'un message SOAP
  - Enveloppe **Envelope;** Entête **Header;**
    Corps **Body**

Exemples de documents XML

```
<document/>
```

```
<document> </document>
```

```
<document> Bonjour! </document>
```

```
<document>
  <salutation> Bonjour! </salutation>
</document>
```

```
<?xml version="1.0" standalone="yes" ?>
<document>
  <salutation> Bonjour! </salutation>
</document>
```

Structure d'un élément (XML 1.0)

- Un élément est de la forme:

  `<nom attr='valeur'> contenu </nom>`

- `<nom>` est la *balise d'ouverture*
- `</nom>` est la *balise de fermeture*
- [éléments vides, indifféremment `<nom> </nom>` ou `</nom>`]
- `contenu` est le contenu d'un élément ☺
  - composé d'une liste (peut-être vide) de texte, d'autres éléments, d'instructions de traitement et de commentaires
- `attr='valeur'` représente un *ensemble* éventuellement vide d'*attributs*, c'est à dire de paires (nom,valeur). Un élément ne peut posséder qu'un seul attribut de nom donné

Exemples d'éléments

- <a>
- <a>Bonjour comment va?
- <a>……<a>…
- <a>…Bonjour…Salut

- Contenu d'un élément = Forêt d'éléments ou de texte
- Texte UNICODE: peut représenter n'importe quel alphabet (russe, arabe, japonais, chinois …)

Contrainte sur les noms (détail)

- Un nom d'élément ou d'attribut est une suite non vide de caractères pris parmi
  - les *caractères alphanumériques;* le tiret-souligné (*undescore*); le signe *moins;* le *point*; le caractère *deux-points* (:) sens particulier
- qui doit satisfaire les contraintes suivantes
  - le premier caractère doit être alphabétique ou un tiret-souligné
  - les trois premiers caractères ne doivent pas former une chaîne dont la représentation en lettres minuscules est "xml".

Exemples de noms d'éléments	
corrects	incorrects
`_toto`	`1998-catalogue`
`Nom_société`	`XmlSpécification`
`xsl:rule`	`nom société`
`X.11`	

Syntaxe des attributs (XML 1.0)

- Un attribut est une paire `nom='valeur'` qui permet de caractériser un élément. Un élément peut avoir plusieurs attributs. Dans ce cas, les paires `nom='valeur'` seront séparées par un espace.
  - `<rapport langue='fr' dern-modif='08/07/99'>`
  - `<annuaire generator='SQL2XML V2.0' update='07.08.99'>`
- La *valeur* d'un attribut est une chaîne encadrée par des guillemets (`"`) ou des apostrophes simples (`'`). Une valeur d'attribut ne doit pas contenir les caractères `^`, `%` et `&`.
- Un élément a un ensemble d ’attributs (ordre n’a pas de sémantique pour les attributs)

Document *bien formé* (XML 1.0)

- Un document XML doit représenter un *arbre d'éléments*
  - Il existe dans un document un et un seul élément père qui contient tous les autres. C'est la *racine* du document.
  - Un élément distinct de la racine est totalement inclus dans son père
    - `<p> <b> bla bla </p> bla </b> NON!`
- On dit qu'un document XML doit être bien formé

Modèle de données semi-structurées

XML: Format et langage pour les données semi-structurées du Web

- Objectif : un formalisme pour la description et l'échange de données sur le Web,
- Principes de XML
  - balisage structurel (issu de SGML)
  - balisage défini par les auteurs : souplesse
  - séparer la *structure logique* des données de leur *présentation*
  - feuille de style (XSL) : ensemble de règles pour la réalisation sur un médium cible

Avantages

- Serveur de documents XML versatile
  - un seul format pour la majorité des documents
- Interopérabilité des outils
  - format textuel "lisible", indépendant de la plateforme
- Structures plus typées
  - Spécifier des contraintes sur la structure
  - Contrôler la validité d'un document (structure conforme)
- Requêtes sur la structure
  - Critère de recherche plus précis
    - Ex: Trouver les documents dont l'auteur est Victor Hugo
  - Plus précis q'une recherche plein texte
    - Ex: Trouver les documents contenant "Victor Hugo"

Document XML semi-structuré

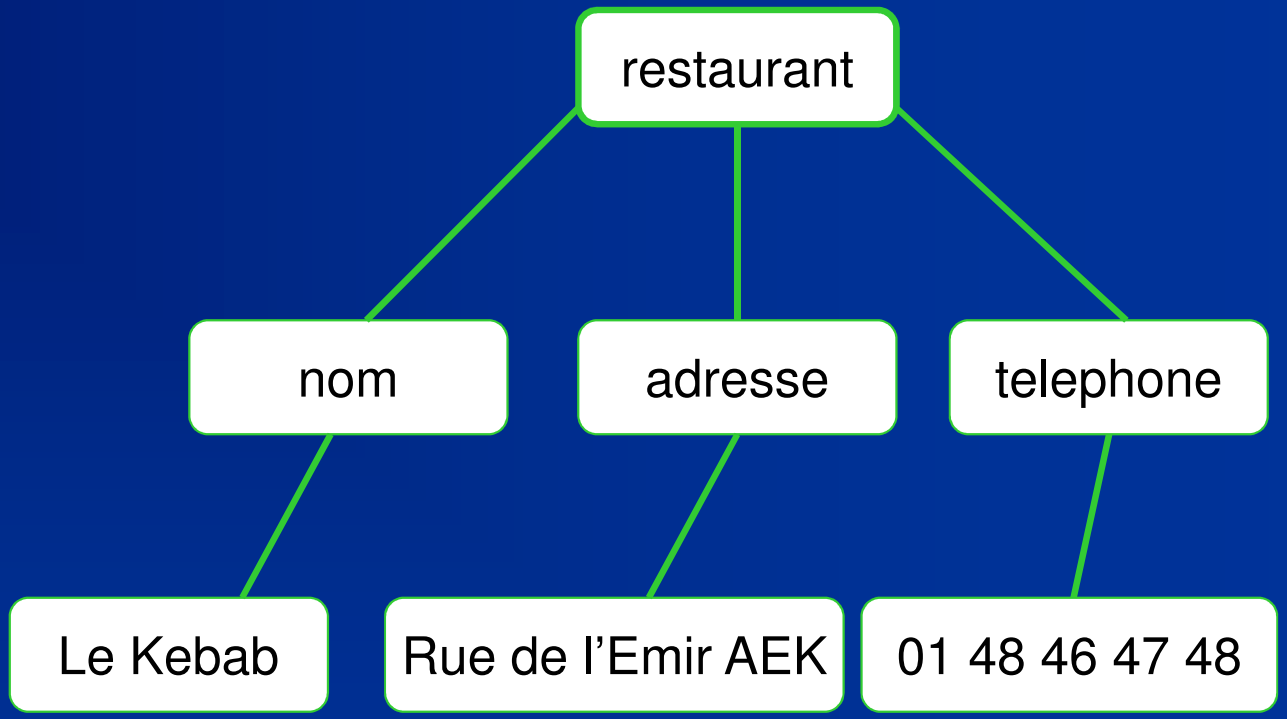
- Structure flexible
  - Irrégulière : intégration de données de structure proche mais différente
  - Structure implicite : les balises sont dans le contenu
  - Structure partiellement définie
  - Structure logique
    - prologue avec définition de la structure (schéma)
    - arbre d'éléments (données)
  - Structure physique
    - document contenant des références à des entités ("macro instructions") : factorisation afin de réduire les répétitions.

Représentation arborescente

- Un document XML peut se représenter sous la forme arborescente
  - Met en évidence la structure hiérarchique du document
  - Facilite la conception des traitements
    - permet de spécifier des manipulations de données XML
    - utilisé par les applications qui gèrent les documents en mémoire
      - Ex: éditeur XML

Exemple

```
<restaurant>
    <nom>Le Kebab </nom>
    <adresse>
        Rue de l'Emir AEK
    </adresse>
    <telephone>
        01 48 46 47 48
    </telephone>
</restaurant>
```



Le modèle en arbre des documents est spécifié par le *Document Object Model* (DOM)