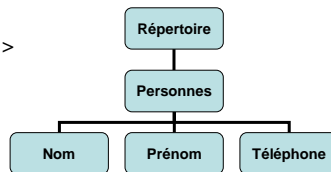


Exemple : Un document XML simple

```
<?xml version="1.0"?>
<Répertoire Type="Personnel">
  <Personne>
    <Nom>Dupond</Nom>
    <Prénom>Jean</Prénom>
    <Téléphone>0160257912</Téléphone>
  </Personne>
  <Personne>
    <Nom>Durand</Nom>
    <Prénom>Jeanne</Prénom>
    <Téléphone/>
  </Personne>
</Répertoire>
```





XML est constitué physiquement par des caractères alphanumériques

Le fichier « bonjour.xml » contenant le texte

Bonjour monde !

Chargé dans IE donnera :

**Non valide au niveau
supérieur du document.
Ligne 1, Position 1**
Bonjour monde !



Les caractères doivent être « rangés » dans des balises

Le fichier « bonjour.xml » contenant le texte

<demo>Bonjour monde !

Chargé dans IE donnera :

**La page XML ne peut pas être affichée
Les balises suivantes n'ont pas été fermées : demo.**

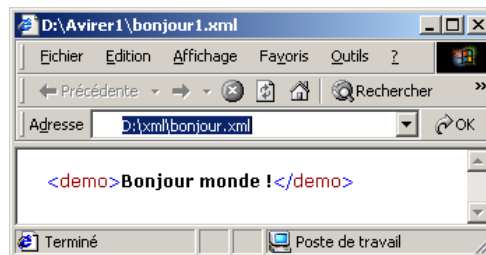
Les caractères doivent être « rangés » dans des balises
ENCADRANTES



Le fichier « bonjour.xml » contenant le texte

`<demo>Bonjour monde !</demo>`

Chargé dans IE donnera :



La balise `<demo>` n'est pas du HTML



XML

=

eXtensible Markup Language



A quoi ça sert ???

Recevoir des données venant d'un outil différent

->Depuis un PC vers un Tél. WAP...

Echanger des données entre applications différentes

->Depuis Excel vers Oracle...

Echanger entre systèmes hétérogènes

->Windows vers Unix, Mac

Structurer des données dans des pages HTML

->Aide aux moteurs de recherche/ extraction de données ...



LES ATOUTS (1)

La lisibilité : aucune connaissance ne doit
théoriquement être nécessaire pour
comprendre un contenu d'un document XML

LES ATOUTS (2)



Une structure arborescente (donc logique)
permettant une manipulation via des objets

LES ATOUTS (3)



Déployable et universel : il peut être facilement distribué
par n'importe quel protocole à même de transporter du
texte, comme HTTP, et dans n'importe quel jeux de
caractères



LES ATOUTS (4)



Normalisé : un document XML est utilisable par toute application pourvue d'un parser (c'est-à-dire un logiciel permettant d'analyser un code XML)



DEFINITIONS



Elément

Chaque document contient un ou plusieurs éléments, leurs limites sont soit délimitées par des balises de début et de fin, ou, pour les éléments vides, par une balise d'élément vide. Chaque élément peut avoir un ou plusieurs attributs.

Exemple (1/5)

<elevage>JP</elevage>

Exemple (2/5)

<elevage ></elevage>

Ou

<elevage/>

Exemple (3/5)

```
<elevage nom= "vancouver" couleur="bringé">JP</elevage>
```

Exemple (4/5)

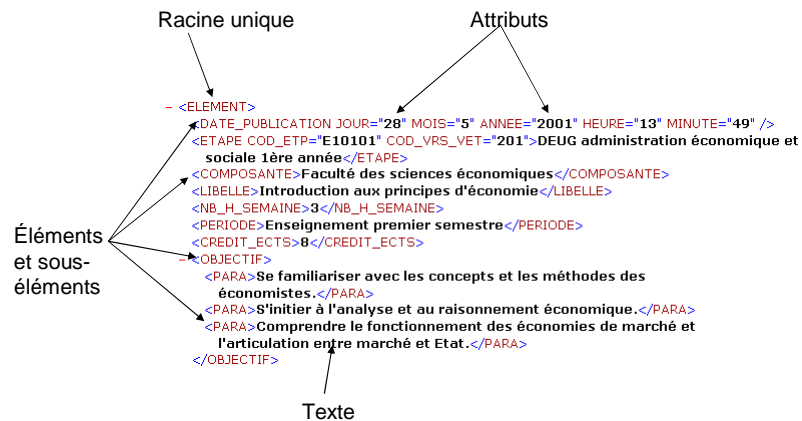


```
<?xml version="1.0" ?>
<infos_PC>
  <description>
    <nom>PC Home</nom>
    <prix_unitaire>5999</prix_unitaire>
    <fournisseur>Sulyo</fournisseur>
    <particularites>Base + lecteur DVD </particularites>
  </description>
</infos_PC>
```


Exemple (5/5)



- XML est une structure arborescente
 - Les nœuds de l'arbre contiennent les données



Document XML BIEN FORMÉ



Un document BIEN FORMÉ obéit aux règles syntaxiques du XML
<http://www.w3.org/TR/REC-xml#dt-wellformed>

En résumé :

- Ne possède qu'une racine
- Tous les éléments doivent être fermés
- Tous les éléments doivent être imbriqués

XML

```
<?xml version="1.0" standalone="yes" ?>
<state stateid="MN">
  <city cityid="12">
    <name>Johnson</name>
    <population>5000</population>
  </city>
  <city cityid="15">
    <name>Pineville</name>
    <population>60000</population>
  </city>
</state>
```

HTML

```
<html>
<h1 id="MN">State</h1>
<h2 id="12">City</h2>
<dl>
  <dt>Name</dt>
  <dd>Johnson</dd>
  <dt>Population</dt>
  <dd>5000</dd>
</dl>
<h2 id="15">City</h2>
<dl>
  <dt>Name</dt>
  <dd>Pineville</dd>
  <dt>Population</dt>
  <dd>60000</dd>
</dl>
</html>
```



... Vers le XHTML ?

Document XML VALIDE

Un document est dit VALIDE lorsque ses éléments sont conformes à une spécification c'est à dire lorsque sa structure est décrite avec précision

C'est le rôle de la DTD

(sera l'objet d'une séance spécifique)



