eXtensible Markup Language

- L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbre, texte riche…) entre systèmes d'informations hétérogènes (interopérabilité).
- Avec ses outils et langages associés une application XML respecte généralement certains principes :
  - la structure d'un document XML est définie et validable par un schéma,
  - un document XML est entièrement transformable dans un autre document XML.

X comme extensible

- HTML
  - Nombre fini de balises
  - Balises pour formater
- XML
  - Possibilité de définir ses propres balises
  - Balises pour structurer
  - DTD

Exemple de document XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Commentaire -->
<ex:collection xml:lang="fr"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:ex="http://exemple.org" >
 <élément>Texte</élément>
 <dc:title>Astérix le Gaulois</dc:title>
 <ex:livre attribut="valeur" type="BD">
   <dc:title>Astérix chez les Belges</dc:title>
   <dc:creator>René Goscinny</dc:creator>
   <dc:creator>Albert Uderzo</dc:creator>
   <dc:description>
      <b>Astérix chez les Belges</b> est un album
      de bande dessinée de la série Astérix le Gaulois
      créée par René Goscinny et Albert Uderzo.
      <br/>
      Cet album publié en 1979 est le dernier de la
      série écrit par René Goscinny.
   </dc:description>
 </ex:livre>
</ex:collection>
```

Structure d'un document XML

- Un document XML a toujours une et une seule racine, le nœud document. La racine peut éventuellement comporter des enfants de type commentaire ou instruction de traitement, elle doit obligatoirement comporter un et un seul élément.
- Un élément est un nœud, désigné par un nom qualifié au sein d'un espace de noms (<espace:élément/>), pouvant contenir la plupart des autres nœuds : texte, éléments, attributs… (à l'exception du nœud document).

Particularités de XML

- répétable
  - Une même propriété peut être répétée (<dc:creator>)
- ordonné
  - L'ordre des éléments est conservé
    (distinguer le premier auteur du second)
- hiérarchique
  - Les éléments XML sont imbricables. Ceci rend ce format particulièrement adapté à représenter des arbres.
- mélangeable
  - XML est plus qu'un format de données, c'est un format de document, permettant de mélanger du texte et des éléments.
- qualifié
  - La qualification des noms contribue à la précision sémantique des contenus balisés.

Prologue

- Une déclaration XML
  ```
  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
  ```
- Instructions de traitement (Processing Instruction)
  Une indication de traitement est destinée aux applications qui manipulent les documents XML
- Une déclaration de type de document
  indique le type de document auquel se conforme le document en question (ex. DTD)
  ```
  <!DOCTYPE rapport SYSTEM "rapport.dtd">
  ```

Elément

- Composant de base
- Identifié par un nom
- Délimité par une balise ouvrante et une balise fermante à ce nom

```
<AUTEUR>Victor Hugo</AUTEUR>
```

- Ou élément vide

```
<PHOTO source="HugoV.gif"/>
```

- Contenu textuel, éléments ou mixte des 2

Attributs

- Inclus dans la balise ouvrante d'un élément
- Composés d'un nom et d'une valeur

```
<AUTEUR NE="1802" MORT="1885">
  Victor Hugo
</AUTEUR>
```

Données

- Constituées d'un flot de caractères
  - Tous les caractères sont acceptés sauf & et <
  - Exemple `<auteurs>Victor Hugo<auteurs>`
- Pour utiliser des caractères spéciaux, une section littérale ou CDATA

```
<![CDATA[<auteurs>S. Fleury &amp; al.</auteurs>]]>
```

SAX et DOM

- SAX (Simple API for XML), qui apparaît en 1998, est une interface de programmation permettant de lire et de traiter des flux XML.
- DOM (Document Object Model) charge l'intégralité d'un document XML dans une structure de données qui peut alors être manipulée puis reconvertie.

SAX : paquetages

- Les principaux composants java sont fournis dans 3 paquetages :

org.xml.sax	*Attributes*
	InputSource
	SAXException
	XMLReader
org.xml.sax.helpers	DefaultHandler
javax.xml.parsers	SAXParser
	SAXParserFactory

SAX : analyse d'un flux

- SAX est événementiel :
  - une balise ouvrante est un événement
  - une balise fermante un autre événement
- L'analyseur de flux, qui hérite de **`DefaultHandler`**, redéfinit :
  - void startDocument()
  - void startElement(String, String, String, Attributes)
  - void characters(char[], int , int)
  - void endElement(String, String, String)
  - void endDocument()

SAX : exemple

```
public class ExempleLectureXML {
  public static void main(String args[]) throws Exception {
    SAXParserFactory spf = SAXParserFactory.newInstance();
    spf.setNamespaceAware(true);
    SAXParser sp = spf.newSAXParser();
    XMLReader parser = sp.getXMLReader();
    Handler handler = new Handler();
    parser.setContentHandler(handler);
    parser.parse("unDocument.xml");
  }
}
```

SAX : exemple.

```
public class Handler extends DefaultHandler {
  public void startDocument() { ... }
  public void startElement(String namespaceURI,
                           String localName,
                           String qualifiedName,
                           Attributes attributs)
  throws SAXException { ... }
  public void characters(char ch[], int start, int length)
  throws SAXException { ... }
  public void endElement(String namespaceURI,
                         String localName,
                         String qualifiedName)
  throws SAXException { ... }
  public void endDocument() { ... }
}
```

SAX : exemple..

```
public class Handler extends DefaultHandler {
  ...
  public void startElement(String namespaceURI,
                           String localName,
                           String qualifiedName,
                           Attributes attributs)
  throws SAXException {
    System.out.println("début: " + qualifiedName);
    for (int index = 0; index < attributs.getLength(); index++) {
      System.out.print("     - " +  attributs.getLocalName(index));
      System.out.println(" + " = " + attributs.getValue(index));
    }
  }
  ...
}
```

SAX : exemple...

```
public class Handler extends DefaultHandler {
  ...
  public void characters(char ch[], int start, int length)
  throws SAXException {
    System.out.println("valeur (car " + start + " à " +
                       (start + length - 1) +  " ): " +
                       new String(ch, start, length));
  }
  public void endElement(String namespaceURI, String localName,
                         String qualifiedName)
  throws SAXException {
    System.out.println("fin:  /" + qualifiedName);
  }
  ...
}
```

JDOM : création d'arborescence

```
Element racine;
org.jdom.Document document;
racine = new Element("personnes");
document = new Document(racine);
Element etudiant = new Element("etudiant");
racine.addContent(etudiant);
Attribute classe = new Attribute(
  "classe", "Licence"
);
etudiant.setAttribute(classe);
Element nom = new Element("nom");
nom.setText("Daniel");
etudiant.addContent(nom);
```

JDOM : affichage enregistrement

```
XMLOutputter sortie = new XMLOutputter(
  Format.getPrettyFormat()
);
sortie.output(document,
 System.out
);

XMLOutputter sortie = new XMLOutputter(
  Format.getPrettyFormat()
);
sortie.output(document,
  new FileOutputStream("nomDeFichier")
);
```

JDOM : lecture et parcours

```
org.jdom.Document document;
Element racine;
SAXBuilder sxb = new SAXBuilder();
document = sxb.build(new File("personnes.xml"));
racine = document.getRootElement();

List listElements = racine.getChildren("etudiant");
Iterator i = listElements.iterator();
while(i.hasNext()) {
  Element courant = (Element)i.next();
  System.out.println(courant.getChild("nom").getText());
}
```

JDOM : Filtrage

```
Filter filtre = new Filter() {
  public boolean matches(Object ob) {
    if(!(ob instanceof Element)) return false;
    Element element = (Element)ob;
    if(element.getChild("classe").getTextTrim()
      .equals("licence")) return(true);
    return false;
  }
};
List resultat = racine.getContent(filtre);
Iterator i = resultat.iterator();
while(i.hasNext()) {
  Element courant = (Element)i.next();
  System.out.println(courant.getAttributeValue("nom"));
}
```

JDOM : modifier une arboresc.

- addContent
- removeAttribute
- removeChild
- removeChildren
- setAttribute
- setContent
- setName
- setText