

University of Jijel
Faculty of Exact Sciences and Computer Science
Department of Computer Science
L3 – Computer Systems

Semi-Structured Data

Chapter 3

XML Kernel

Tarek Boutefara
t_boutefara@univ-jijel.dz
2025/2026

Content

- Introduction to XML
- Structure XML de base
- Domaines nominaux
- Schémas XML

Content

- **Introduction to XML**
- Basic XML Structure
- Nominal Domains
- XML Schemas

Introduction to XML

- Spécification
 - <http://www.w3.org/TR/REC-XML>

Introduction to XML

- Specification
 - <http://www.w3.org/TR/REC-XML>
- The World Wide Web Consortium (W3C)
 - An international community working to develop the web
 - Directed by Tim Berners-Lee
 - 440 members (Adobe, Amazon, Cisco, Facebook, Google, Microsoft, ...)

Introduction to XML

- Specification
 - <http://www.w3.org/TR/REC-XML>
- The World Wide Web Consortium (W3C)
 - Process
 - Discussion of interesting topics,
 - Development of an initial proposal,
 - Participation of working groups,
 - Development of specifications (and guidelines).

Introduction to XML

- Specification
 - <http://www.w3.org/TR/REC-XML>
- The Extensible Markup Language (XML) is a subset of SGML that is completely described in this document. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML.
- XML has been designed for ease of implementation and for interoperability with both SGML and HTML.

Introduction to XML

- Objectives:
 - XML must be directly usable on the Internet,
 - XML must support a wide variety of applications,
 - XML must be compatible with SGML,
 - Programs that read XML must be easy to write,
 - Optional elements must be kept to a minimum.

Introduction to XML

- Objectives:
 - XML documents must be human-readable,
 - XML document design must be easy,
 - XML document design must be formal,
 - XML documents must be easy to create.

Introduction to XML

- Why XML?
 - To define custom and adapted exchange languages,
 - To model data,
 - Document Type Definitions (DTDs),
 - XML Schemas,
 - To publish information,
 - CSS formatting,
 - XSL transformations,
 - To archive data,
 - XML databases

Content

- Introduction to XML
- **Basic XML Structure**
- Nominal Domains
- XML Schemas

Basic XML Structure

- `<?xml version="1.0" encoding="ISO-8859-1" ?>`
- `<book lang="fr">`
 - `<title>Java</title>`
 - `<year>2015</year>`
 - `<price>3500</price>`
- `</book>`

Basic XML Structure

- `<?xml version="1.0" encoding="ISO-8859-1" ?>`
 - Precision of the XML version used in the document

Basic XML Structure

- `<book> ... </book>`
 - The document root

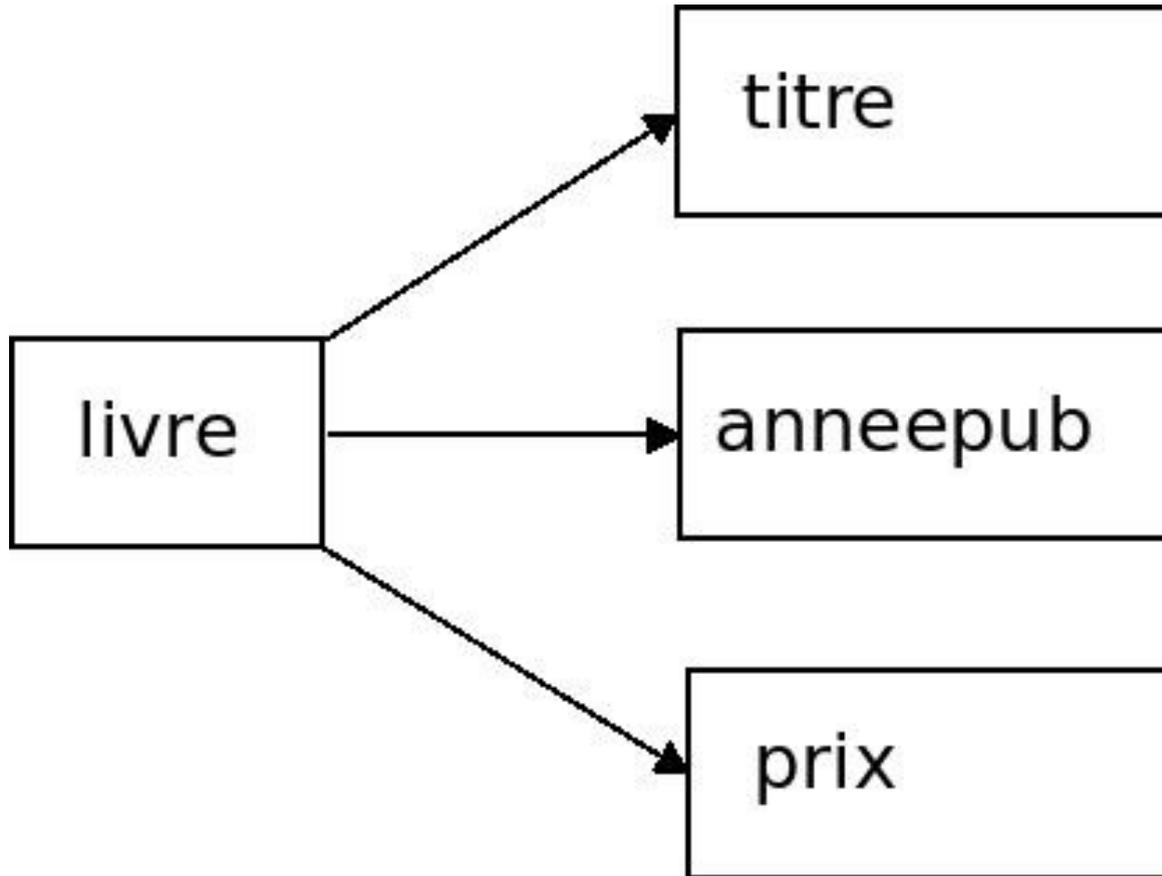
Basic XML Structure

- `<book> ... </book>`
 - `lang= "fr"`
 - Attribut for element `<book>`

Basic XML Structure

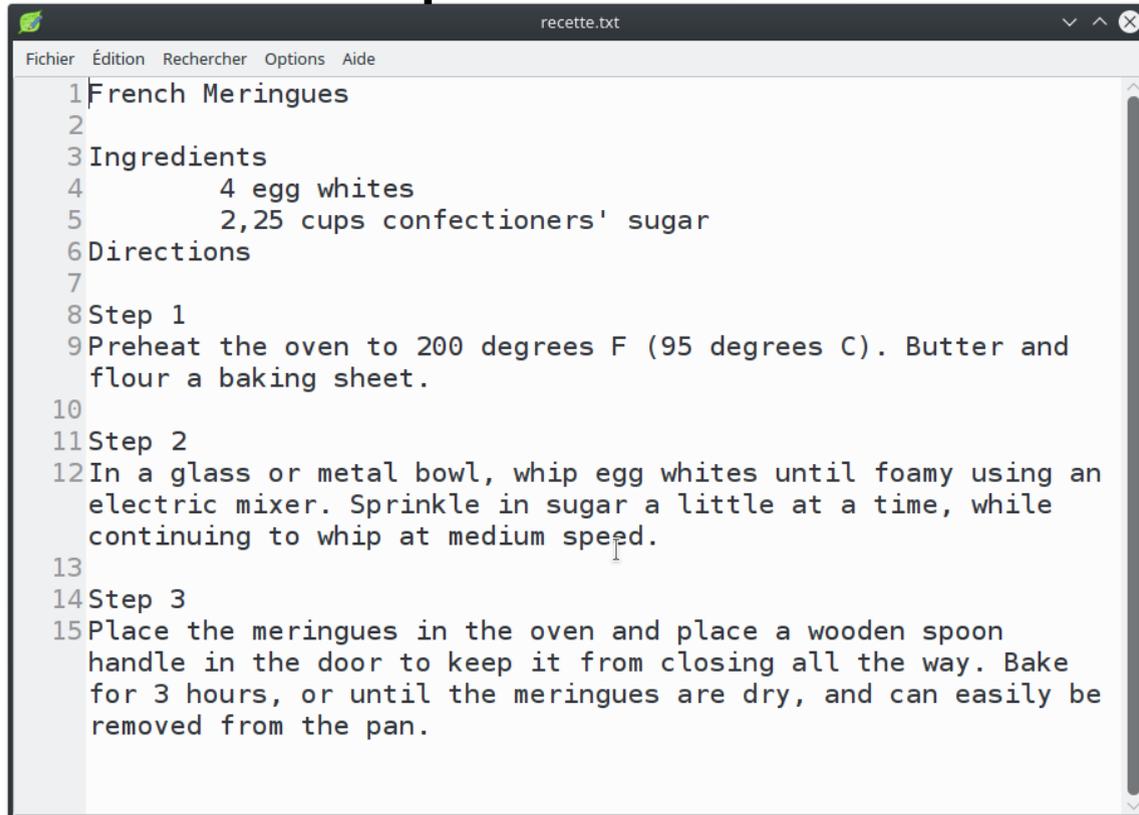
- The elements:
 - `<title>Java</title>`
 - `<year>2015</year>`
 - `<price>3500</price>`
- Three sub-elements from the element `<book>`

Basic XML Structure



Basic XML Structure

- A second example



The image shows a screenshot of a text editor window titled "recette.txt". The window has a menu bar with "Fichier", "Édition", "Rechercher", "Options", and "Aide". The text content is as follows:

```
1 French Meringues
2
3 Ingredients
4     4 egg whites
5     2,25 cups confectioners' sugar
6 Directions
7
8 Step 1
9 Preheat the oven to 200 degrees F (95 degrees C). Butter and
  flour a baking sheet.
10
11 Step 2
12 In a glass or metal bowl, whip egg whites until foamy using an
  electric mixer. Sprinkle in sugar a little at a time, while
  continuing to whip at medium speed.
13
14 Step 3
15 Place the meringues in the oven and place a wooden spoon
  handle in the door to keep it from closing all the way. Bake
  for 3 hours, or until the meringues are dry, and can easily be
  removed from the pan.
```



/home/tarek/Enseignement/XML X +



Most Visited Linux Mint Blog Forums Community News Bookmark this page Other Bookmarks

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-<recette id="1336">
  <titre>French Meringues</titre>
  -<ingredients>
    <ingredient qte="4">egg whites</ingredient>
    <ingredient qte="2.25" unite="cup">confectioners' sugar</ingredient>
  </ingredients>
  -<directions>
    -<step n="1">
      Preheat the oven to 200 degrees F (95 degrees C). Butter and flour a baking sheet.
    </step>
    -<step n="2">
      In a glass or metal bowl, whip egg whites until foamy using an electric mixer. Sprinkle in sugar a little
      at a time, while continuing to whip at medium speed.
    </step>
    -<step n="3">
      Place the meringues in the oven and place a wooden spoon handle in the door to keep it from closing
      all the way. Bake for 3 hours, or until the meringues are dry, and can easily be removed from the pan.
    </step>
  </directions>
</recette>
```

Basic XML Structure

- A second example
 - Visualization:
 - Several tools exist for tree-like visualization,
 - but far fewer for graph-like visualization,
 - In this example:
 - <https://codebeautify.org/xmlviewer>

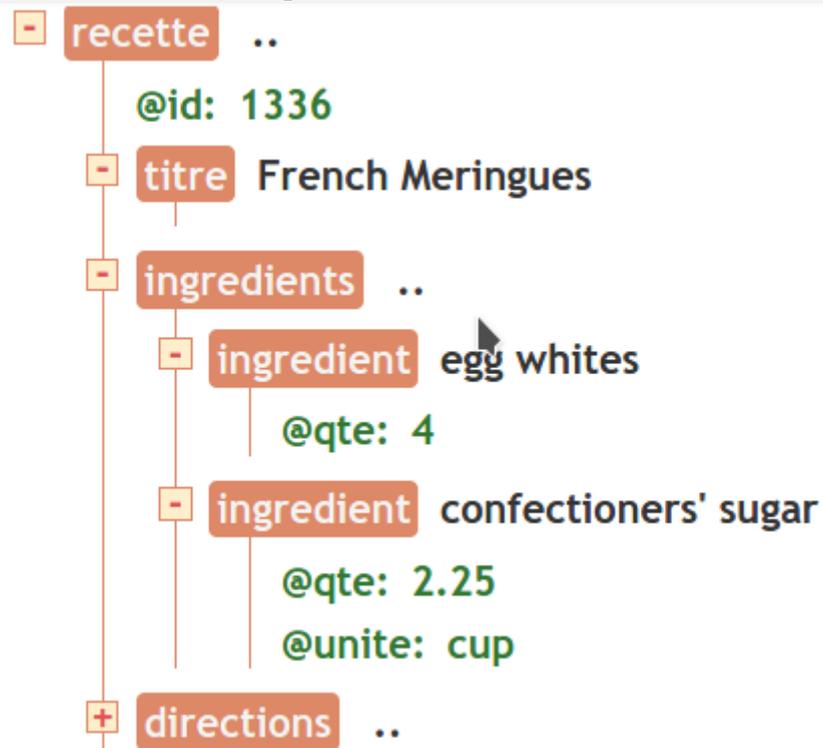
Basic XML Structure

- A second example



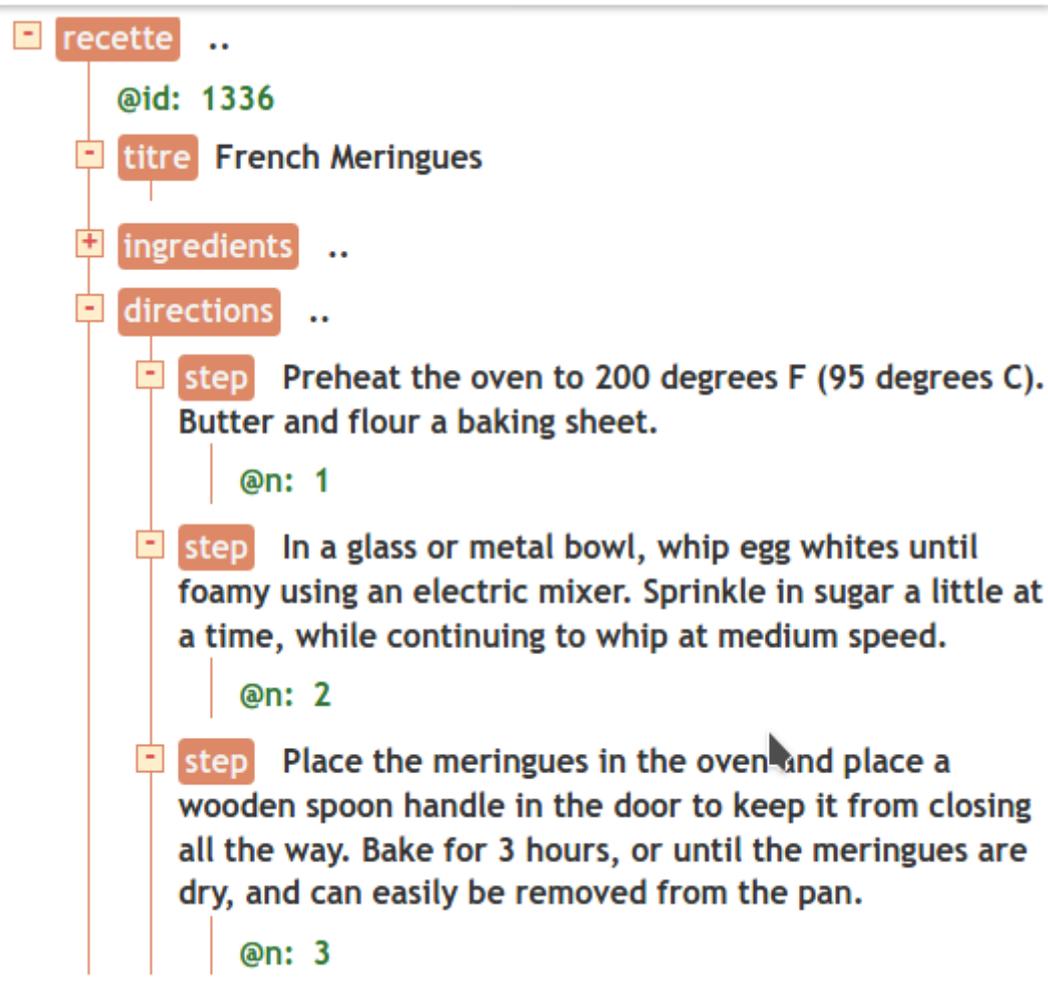
Basic XML Structure

- A second example



Basic XML Structure

- A sec



Content

- Introduction to XML
- Basic XML Structure
- **Nominal Domains**
- XML Schemas

Nominal Domains

- Freedom in defining tags:
 - Risks of conflicts during merging,
- Example:
 - Book
 - Client
 - Order (?)

Nominal Domains

```
<?xml version="1.0" ?>
<book>
  <title>Java</title>
  <authors>
    <author>
      <name>Jamie Chan</name>
      <nationality>American</nationality>
    </author>
  </authors>
  <price>11.99</price>
</book>
```

Nominal Domains

```
<?xml version="1.0" ?>  
<client>  
  <code>1226</code>  
  <name>Benmohammed Mohammed</name>  
  <address>Jijel</address>  
</client>
```

Nominal Domains

```
<?xml version="1.0" ?>
<order>
  <client>
    <code>1226</code>
    <name>Benmohammed Mohammed</name>
    <address>Jijel</address>
  </client>
  <books>
    <book>
      <title>Java</title>
      <authors>
        <author>
          <name>Jamie Chan</name>
          <nationality>American</nationality>
        </author>
      </authors>
      <price>11.99</price>
    </book>
  </books>
</order>
```

Nominal Domains

- Namespaces
 - Defining name collections,
 - Applicable to tags and attributes
 - Each collection is defined by a URI,
 - The URI is associated with a prefix
 - `xmlns:prefix="URI"`
 - The prefix must be declared by an ancestor

Nominal Domains

```
<?xml version="1.0" ?>
<order
  xmlns="http://www.univ-jijel.dz/orders/"
  xmlns:l="http://www.univ-jijel.dz/books/"
  xmlns:c="http://www.univ-jijel.dz/clients/">
  <c:client>
    <c:code>1226</c:code>
    <c:name>Benmohammed Mohammed</c:name>
    <c:address>Jijel</c:address>
  </c:client>
  <books>
    <l:book>
      <l:title>Java</l:title>
      <l:authors>
        <l:author>
          <l:name>Jamie Chan</l:name>
          <l:nationality>American</l:nationality>
        </l:author>
      </l:authors>
      <l:price>11.99</l:price>
    </l:book>
  </books>
</order>
```

Content

- Introduction to XML
- Basic XML Structure
- Nominal Domains
- **XML Schemas**

XML Schemas

- Why?
 - Allowed elements
 - Defining the hierarchy
 - Allowed attributes
 - Attribute value model

DTD

- 5 "components" to define:
 - Element,
 - Attribute,
 - Entities
 - The set of special characters
 - PCDATA
 - parsed character data: will be parsed
 - CDATA
 - character data: will not be parsed

DTD

- Element
 - Declaration
 - `<!ELEMENT element_name category>`
 - `<!ELEMENT element_name (sub_elements)>`

DTD

- Element
 - Operators:
 - +: minimum one,
 - *: zero or more,
 - ?: zero or one,
 - ,: list of elements,
 - |: OR,
 - #PCDATA
 - EMPTY (without parentheses)
 - ANY (without parentheses)

DTD

- Attributes
 - Declaration
 - `<!ATTLIST element_name attribute_name attribute_type attribute_value>`

DTD

- Attributes
 - Attribute Types
 - CDATA: string,
 - (val1|val2|val3): a value from the list,
 - ID: unique value,
 - ...

DTD

- Attributes
 - Values
 - "value": default value,
 - #REQUIRED: mandatory,
 - #IMPLIED: optional,
 - #FIXED "value": fixed value.

DTD

- Example:
 - Recipe:

```
<!DOCTYPE recipe [  
  <!ELEMENT recipe (title,ingredients,directions)>  
  <!ELEMENT title (#PCDATA)>  
  <!ELEMENT ingredients (ingredient+)>  
  <!ELEMENT ingredient (#PCDATA)>  
  <!ELEMENT directions (step+)>  
  <!ELEMENT step (#PCDATA)>  
  
  <!ATTLIST recipe id CDATA #REQUIRED>  
  <!ATTLIST ingredient qty CDATA #REQUIRED>  
  <!ATTLIST ingredient unit CDATA #IMPLIED>  
  <!ATTLIST step n CDATA #REQUIRED>  
  

```

DTD

- Declaration
 - Start of document
 - In a separate file
 - `<!DOCTYPE recipe SYSTEM "recipe.dtd">`

DTD

- Example in Java:
 - Step 01: You need to create an ErrorHandler:
 - ErrorHandler handler = new ErrorHandler() {
 - public void error(SAXParseException exception) throws SAXException { }
 - public void fatalError(SAXParseException exception) throws SAXException { }
 - public void warning(SAXParseException exception) throws SAXException { }
 - });

DTD

- Example in Java:
 - Step 2: setValidating(true) and we provide the handler:
 - DocumentBuilderFactory domFactory = DocumentBuilderFactory.newInstance();
 - domFactory.setValidating(true);
 - DocumentBuilder builder = domFactory.newDocumentBuilder();
 - builder.setErrorHandler(**handler**) ;
 - Document doc = builder.parse("employee.xml");

XSD : XML Schema

- The schema itself is an XML file,
- It allows you to define more complex types,
- It allows you to specify data types:
 - Integer,
 - Date,
 - ...

XSD : XML Schema

- Exemple : Recette

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="recipe">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="title"/>
        <xs:element name="ingredients">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ingredient" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:float" name="qty" use="optional"/>
                      <xs:attribute type="xs:string" name="unit" use="optional"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

XSD : XML Schema

- Exemple : Recette (suite)

```
<xs:element name="directions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="step" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute type="xs:byte" name="n" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute type="xs:short" name="id"/>
</xs:element>
</xs:schema>
```

University of Jijel
Faculty of Exact Sciences and Computer Science
Department of Computer Science
L3 – Computer Systems

Semi-Structured Data

Chapter 3

XML Kernel

Tarek Boutefara
t_boutefara@univ-jijel.dz
2025/2026