



XML

eXtensible Markup Language

langage de balisage extensible

1



Concours d'acronymes?

- XML **eXtensible Markup Language**
- W3C **World Wide Web Consortium**
- DTD **Définition de Type de Document**
- XSD **XML Schema Document**
- XPATH **XML Path**
- XSLT **eXtensible Stylesheet Language Transformations**
- CSS **Cascading Style Sheet**
- URI **Universal Ressource Information**
- XSL-FO **eXtensible Style Language Formatting Objects**
- XLINK **XML Link**
- XPOINTER **XML Pointer**
- API **Application Programming Interface**
- DOM **Document Object Model**
- IDL **Interface Definition Language**

2



XML : la famille

- Né : fin 96
- Père : W3C
- Petit-fils de SGML (ISO-1986)
- Cousin d'HTML
- Reconnu le : 10/02/98 – version 1.0
- Descendance – XHTML, MathML, SVG, ...

3



X comme eXtensible

- HTML : nombre fini de balises
- XML : possibilité de définir les balises
- HTML : balises pour la mise en page
- XML : balises pour structurer les données
- DTD ou Schéma pour définir les balises (et leur grammaire)

4



M comme Markup

- Tous les éléments sont repérés au moyen de balises
- Les balises servent à structurer le document

5



L comme Language

- Langage de description de documents
- Règles à respecter

6



Le document XML

- Données binaires
- Données textuelles
- Norme Unicode
- Marques :
 - Balises de début, de fin, références, commentaires, instructions de traitement
- Validation par une « Document Type Definition » (DTD) ou un « XML Schema »

7



Règles syntaxiques pour avoir un document xml dit « bien formé »

1. Commencer par une déclaration XML (prologue)
2. Balisage sensible à la casse
3. La valeur des attributs doit être quotée
4. Balises non vides appariées `
</br>`
5. Balises vides fermées `
`
6. Les éléments ne doivent pas se chevaucher
`<mois><jour> </mois></jour>` **interdit**
7. Un élément (racine) doit encapsuler tous les autres
8. Ne pas utiliser les caractères `<` et `&` seuls

8



Élément

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

`<AUTEUR> Victor Hugo </AUTEUR>`

- Ou élément vide (ou balise auto-fermée)

`<PHOTO Source="victor.gif" />`

- Contenu textuel, éléments ou mixte

9



Les attributs

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

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

10



...et aussi

- `<!-- des commentaires -->`
- Des entités externes/internes, analysables ou non
`&`; `"`; `>`; `<`;
- `<![CDATA [`
...tout et n'importe quoi...
...0x01265423deadbeef49653453462...
`]]>`

11



Exemple de document XML

```
<?xml version="1.0" encoding="ISO-8859-1"
?>
<memo langue="fr" urgence="maximale">
  <to>Enfants Sages</to>
  <from>Père Noël</from>
  <date>7/12/2017</date>
  <body>Livraison confirmée
    <jour>25/12</jour>
    <heure>0h00</heure>
    <lieu>cheminée</lieu>
  </body>
</memo>
```

12

Aparté: autre langage pour transporter l'information JSON: JavaScript Object Notation

```
<?xml version="1.0" encoding="ISO-8859-1"
?>
<memo langue="fr" urgence="maximale">
  <to>Enfants Sages</to>
  <from>Père Noël</from>
  <date>7/12/2017</date>
  <body>Livraison confirmée
    <jour>25/12</jour>
    <heure>0h00</heure>
    <lieu>cheminée</lieu>
  </body>
</memo>
```

```
{
  "memo": {
    "-langue": "fr",
    "-urgence": "maximale",
    "to": "Enfants Sages",
    "from": "Père Noël",
    "date": "7/12/2017",
    "body": {
      "#text": "Livraison confirmée",
      "jour": "25/12",
      "heure": "0h00",
      "lieu": "cheminée"
    }
  }
}
```

13

XML JSON converter

The screenshot shows a web browser window displaying the 'XML JSON converter' website. The page has a header with the site name 'utilities-online.info' and 'engineered by osys'. Below the header, there are navigation links like 'My IP', 'Temporary', 'J'aime 103', 'Tweet', 'nix E', 'G+', 'Time conv', 'Partages', and '63K bts'. The main content area is titled 'XML to JSON and JSON to XML converter online' and includes a description: 'In this page you can convert an xml to json and viceversa.' There are two text input fields: 'XML' and 'JSON'. The 'XML' field contains the following code:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<memo langue="fr" urgence="maximale">
  <to>Enfants Sages</to>
  <from>Père Noël</from>
  <date>7/12/2017</date>
  <body>Livraison confirmée
    <jour>25/12</jour>
    <heure>0h00</heure>
    <lieu>cheminée</lieu>
  </body>
</memo>
```

 The 'JSON' field contains the following code:

```
{
  "memo": {
    "-langue": "fr",
    "-urgence": "maximale",
    "to": "Enfants Sages",
    "from": "Père Noël",
    "date": "7/12/2017",
    "body": {
      "#text": "Livraison confirmée",
      "jour": "25/12",
      "heure": "0h00",
      "lieu": "cheminée"
    }
  }
}
```

 There are buttons for 'get example xml', 'XML to JSON', and 'JSON to XML'.

14

Autre exemple XML vs JSON

```
<?xml version="1.0" standalone="yes" ?>
<state>
  <stateName>Minnesota</stateName>
  <city>
    <cityName>Johnson</cityName>
    <population>5000</population>
  </city>
  <city>
    <cityName>Pineville</cityName>
    <population>60000</population>
  </city>
</state>
```

```
{
  "state": {
    "stateName": "Minnesota",
    "city": [
      {
        "cityName": "Johnson",
        "population": "5000"
      },
      {
        "cityName": "Pineville",
        "population": "60000"
      }
    ]
  }
}
```

15



Exemple de DTD

```
<?xml version="1.0" Encoding="ISO-8859-1" ? >
<!DOCTYPE memo [
  <!ELEMENT memo (to, from, date, body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT date (#PCDATA)>
  <!ELEMENT body (#PCDATA|jour|heure|lieu)>
    <!ELEMENT jour (#PCDATA)>
    <!ELEMENT heure (#PCDATA)>
    <!ELEMENT lieu (#PCDATA)>
  <!!ATTLIST memo langue CDATA "Alsacien">
  <!!ATTLIST memo urgence CDATA "normal">
]>
```

16

Exemple XML Schema (XSD)



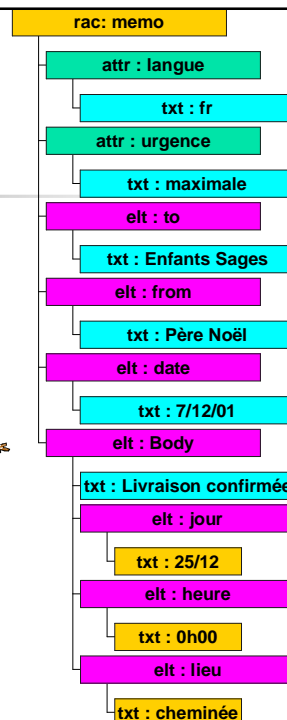
```
<?xml version="1.0" Encoding="ISO-8859-1" ? >
<xsd:element name="memo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="to" type="xsd:string" />
      <xsd:element name="from" type="xsd:string" />
      <xsd:element name="date" type="xsd:date" />
      <xsd:element name="body" type="xsd:string">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="jour" type="xsd:date" />
            <xsd:element name="heure" type="xsd:time" />
            <xsd:element name="lieu" type="xsd:string" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="langue" type="xsd:string" use="optional" />
    <xsd:attribute name="urgence" type="xsd:string" use="required" />
  </xsd:complexType>
</xsd:element>
```

17

Le Document Object Model (Le DOM)



```
<?xml version="1.0" Encoding="ISO-8859-1" ? >
<memo langue="fr" urgence="maximale">
  <to> Enfants Sages </to>
  <from> Père Noël </from>
  <date> 7/12/2017 </date>
  <body> Livraison confirmée
    <jour>25/12</jour>
    <heure>0h00</heure>
    <lieu> cheminée</lieu>
  </body>
</memo>
```



18

Le langage Xpath pour la localisation de noeuds

- Exemple de requête:
`/memo//[lieu="cheminée"]`
- Chemin relatif/absolu
 - `/memo/body/lieu`
- Axe de localisation
 - `self`, `child`, `descendant`, `parent`, `ancestor`, ...
- Test de nœuds
 - `text()`, `comment()`, ...
- Prédicats
 - `last()`, `position()`, `count()`, `name()`, ...

19


Editeur XML

The screenshot displays the Oxygen XML Editor interface. The top menu bar includes File, Edit, Find, Project, DITA Maps, Options, Tools, Author, Document, Window, and Help. The main workspace is divided into several panes:

- Project:** A tree view on the left showing a project named 'sample.xpr' with various files and folders like 'garage', 'it-book', 'learning', 'docbook', 'ooxml', 'tei', 'xhtml', 'xproc', 'ot.xml', 'personal-schema.css', 'personal-schema.xml', 'personal.css', 'personal.dtd', 'personal.xml', and 'personal.xsd'.
- Outline:** A pane below the project tree showing the structure of the selected 'personal.xsd' file, including elements like 'email', 'family', 'given', 'link', 'name', 'person', 'personnel', and 'url'.
- Diagram:** A central pane showing a graphical representation of the XML schema. It includes nodes for 'person' (with a description 'Specifies information about a person.'), 'name' (with a description 'Specifies the person family and given name.'), 'family' (with a description 'The person last name.'), 'given' (with a description 'The person first name.'), 'email' (with a description 'Email address for this person.'), and 'url' (with a description 'Enter an URL for this person.').
- Palettes:** On the right, there are two palettes: 'Basic components' (containing element, group, attribute, attribute group, complex type, simple type, sequence, choice, all, any, any attribute) and 'Compositors and Wildcards' (containing sequence, choice, all, any, any attribute).
- Attributes:** A pane at the bottom right showing the attributes of the selected element, including 'Name', 'Is Reference', 'Type', 'Default', 'Fixed', 'Min Occurs', and 'Max Occurs'.

The main editor area shows the XML document 'personal.xml' with the following content:

```
<?xml version="1.0" encoding="UTF-8" ?>
<personnel>
  <person id="Big.Boss">
    <name>
      <family>Boss</family>
      <given>Big</given>
    </name>
    <email>chief@oxygenxml.com</email>
  </person>
</personnel>
```



Xmlisation d'une page HTML?

Travail à réaliser

22

Soit la page
HTML suivante

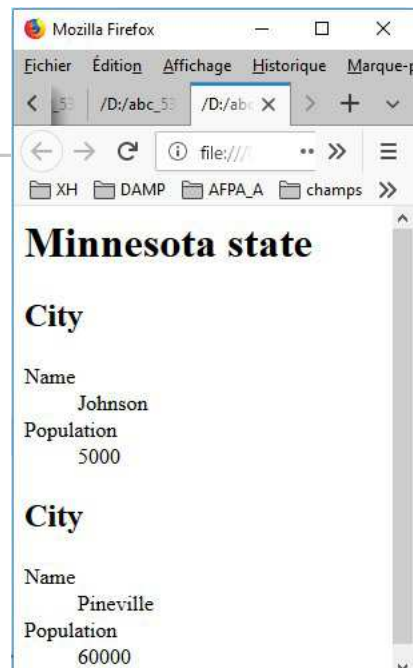


Rappel sur le HTML
sert au navigateur

■ présentation et données

HTML

```
<html>
<h1 id="ST">Minnesota 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>
```



« Xmliser » mon site web

■ Extraire les données

HTML

```
<html>
<h1 id="S">
state</h1>
<h2 id="1">
<dl>
<dt>Name
<dd>John
<dt>Popu
<dd>5000
</dl>
<h2 id="1">
<dl>
<dt>Name
<dd>Pine
<dt>Popu
<dd>6000
</dl>
</html>
```

Minnesota state

Name	Johnson
Population	5000
Name	Pineville
Population	60000

XML

```
<?xml version="1.0" standalone="yes"?>
<state>
  <stateName>Minnesota</stateName>
  <city>
    <cityName>Johnson</cityName>
    <population>5000</population>
  </city>
  <city>
    <cityName>Pineville</cityName>
    <population>60000</population>
  </city>
</state>
```

Débogage du XML grâce à firefox

Erreur d'analyse XML : balise ne correspondant pas. Attendez l'élément de fermeture correspondant.

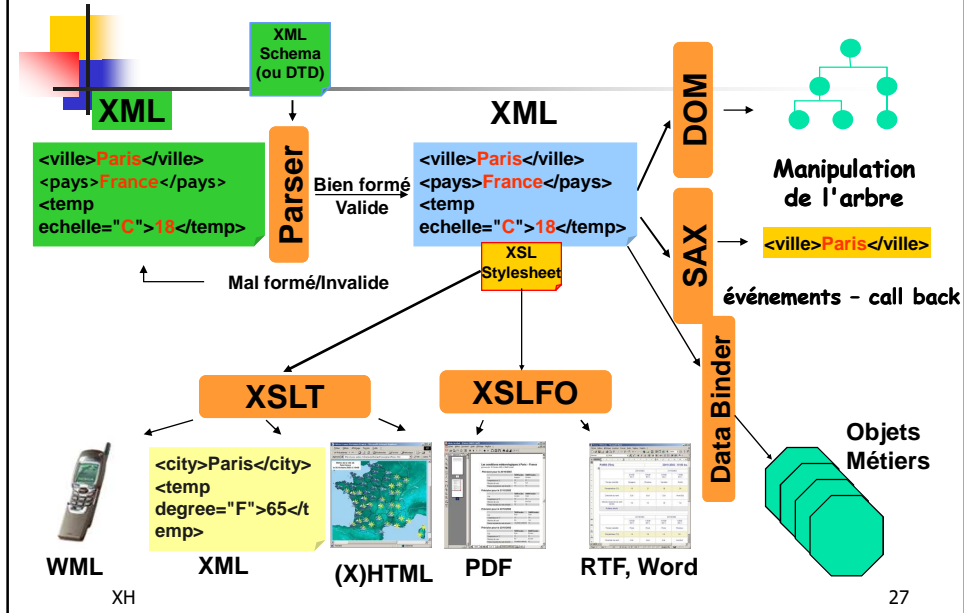
Emplacement : file:///E:/05_LIVRAISON_et_Eval_CDI2017_d...
Numéro de ligne 13, Colonne 4 :

```
</article>
```

Aucune information de style ne semble associée à ce fichier XML. L'arbre du document est affiché ci-dessous.

```
<state>
  <stateName>Minnesota</stateName>
  <city>
    <cityName>Johnson</cityName>
    <population>5000</population>
  </city>
  <city>
    <cityName>Pineville</cityName>
    <population>60000</population>
  </city>
</state>
```

Synthèse des outils pour XML



END

THE END