Langages à balises : une introduction

LIONEL MÉDINI
UFR INFORMATIQUE
UNIVERSITÉ CLAUDE BERNARD LYON 1

D'après le cours de Yannick Prié
2010-2011 – Master SIB
M1 – UE 3 / Bloc 2 – Cours 1

Plan

- **Préambule**
- Introduction
- Balise et balisage
- Structure et arborescence
- Standard Generalized Markup Language (SGML)
- Extensible Markup Language (XML)
- Conclusion

Objectif généraux du module

- Comprendre les grands principes de la représentation de données et de documents numériques à l'aide d'un langage à balises
- Découvrir XML, son histoire et son fonctionnement
- Définir des langages basés sur XML à l'aide de DTD
- Apprendre les bases de XHTML pour la génération de pages web
- S'initier à la transformation de documents en utilisant XSL et un moteur XSLT

Objectifs de ce cours

- Introduction aux langages à balises et à leurs principes
  - langages à balises
    - balises et balisage
    - aperçu de langages
    - présentation de la « galaxie XML »
  - Structuration de documents
    - arbres
  - XML
    - généralités
    - syntaxe
    - synthèse

Généralités sur les 3 cours XML

- Contenus des 3 cours
  - Notions d'arborescence et de balisage
  - Syntaxe XML
  - DTD
  - Schémas
- Objectifs
  - Comprendre la syntaxe XML
  - Savoir construire un document XML en fonction d'une grammaire donnée (DTD ou schéma XML)
  - Créer une grammaire pour un usage particulier

Contenu des 3 cours

- Structuration de documents
  - Balise et balisage
  - Arbre et arborescence
- Documents XML
  - Syntaxe XML et documents bien formés
  - Espaces de noms XML
- Types de documents XML
  - DTD et documents valides
  - Introduction à XML-Schema
- Le monde XML
  - Quelques normes liés à XML
  - Quelques DTD importantes

Plan

- Préambule
- **Introduction**
- Balise et balisage
- Structure et arborescence
- Standard Generalized Markup Language (SGML)
- Extensible Markup Language (XML)
- Conclusion

Donnée, information, document

- Définitions
  - Donnée : valeur associée à un type de données
    - Décrite par des caractéristiques de forme
    - Indépendante de son interprétation : donnée « brute »
  - Élément d'information : ensemble de données faisant « sens »
    - On parle aussi de « grains d'information »
  - Document : regroupement cohérent d'éléments d'information
    - Autour d'une thématique commune
    - Selon une structure donnée

Manipulation de documents numériques

- Document numérique
  - Abstraction / représentation d'un artefact physique
    - Image sur écran, impression
  - Manipulations et gestion à l'aide d'ordinateurs
    - Création, duplication
    - Édition, modification
    - Transformation
- D'où possibilité
  - d'accéder au texte des différentes parties des documents
  - de manipuler la structure des documents
  - de générer plusieurs structures physiques à partir d'une structure logique

Types de structuration

- Définitions
  - Information structurée : bases de données
    - Éléments d'information stockés séparément
    - Accès par requêtes
    - Traitements facilités
  - Information non structurée : corpus documentaires
    - Éléments d'information stockés sous forme de textes
    - Accès par recherche d'information
    - Traitements complexes
  - Information semi-structurée : langages à balises
    - Éléments d'information stockés dans des documents
    - Permet les deux types d'accès et de traitements

Plan

- Préambule
- Introduction
- **Balise et balisage**
- Structure et arborescence
- Standard Generalized Markup Language (SGML)
- Extensible Markup Language (XML)
- Conclusion

Notions de balise et de balisage

- Définitions
  - Balise : signe marquant une position particulière
    - Signe = élément d'information
    - Marquer = permettre la distinction
      - Ex : fusée de détresse, signal radio, « tag » HTML
    - Position = par rapport à l'espace considéré
      - Ex : espace 3D (avion), 2D (bateau), 1D (document)
    - Particulière = en général, la position d'un élément de l'espace qu'on cherche à repérer
      - Ex : aéroport, bateau, élément d'information
  - ⇒ Type d'information facilement repérable permettant d'identifier d'autres éléments informationnels (i.e. « méta-information »)

Notions de balise et de balisage

- Définitions
  - Balisage documentaire : utilisation de balises
    - Pour marquer des points précis d'un document
    - Pour marquer des zones (segments) de document
      - Balisage de début et de fin de zone
    - Pour structurer le document
  - ⇒ Utilisation de plusieurs types de balises
    - ⇒ En fonction du type d'élément à marquer
    - ⇒ En fonction du type de marquage (point, début, fin)

Notion de langage à balises

- Tous les langages ayant pour objectif de représenter de l'information en utilisant des balises
- Définis par
  - vocabulaire
    - noms des éléments
  - grammaire
    - mode d'organisation des éléments
      - des éléments en contiennent d'autres
  - + attributs des éléments
    - un peu plus de structure (voir cours XML)
- Une description
  - ensemble d'éléments organisés dans un fichier
  - contenus terminaux (texte)

Types de langages à balises

- Langages de *balisage procédural*
  - permettent de décrire la mise en forme (formatage) d'un document
  - ex. : PS, RTF, TeX, HTML

Exemple 1/3

- Le langage Postscript

```
%!PS-Adobe-3.0
%%Title: Microsoft Word - Document1
%%Creator: PSCRIPT.DRV Version 4.0
%%CreationDate: 03/02/02 10:47:00
…
%%EndComments
%%BeginProlog
%%BeginProcSet: Pscript_Res_Emul 1.0 0
/defineresource where{pop}{userdict begin/defineresource{userdict/Resources 2
…
}ifelse}bind readonly def end}ifelse
%%EndProcSet
```

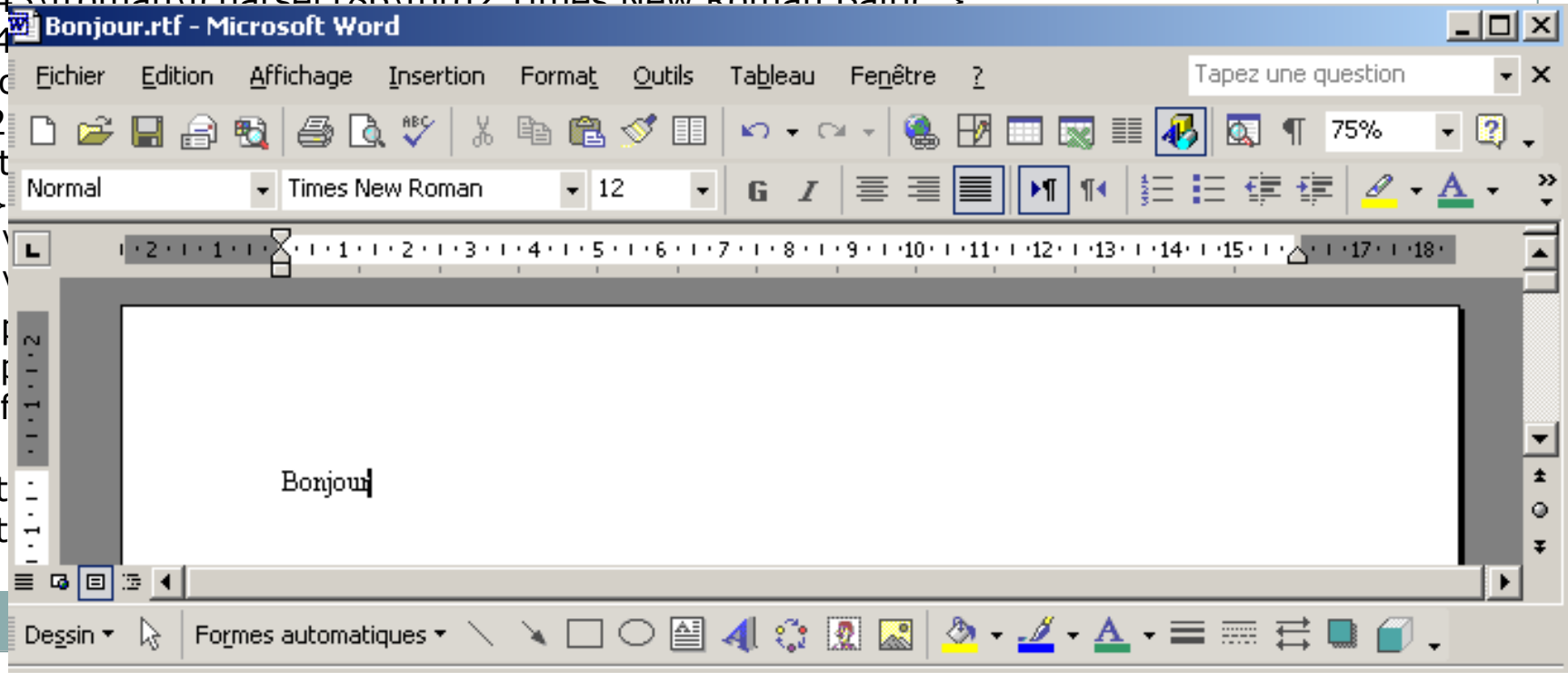
```
%%BeginResource: file
…
%%EndResource
…
%%EndProlog
%%BeginSetup
…
%%BeginFeature: *PageSize A4
…
%%EndFeature
…
%%EndSetup
%%Page: 1 1
%%BeginPageSetup
…
%%EndPageSetup
…
```

```
%%IncludeFont: Courier
…
 (Courier) cvn /Type1
…
(Essai impression)S
…
(%%[ Page: 1 ]%%)  =
%%PageTrailer
%%Trailer
%%DocumentNeededFonts:
%%DocumentSuppliedFonts:
/Pscript_Win_Driver /ProcSet findresource dup /terminate get exec
Pscript_Win_Compat dup /terminate get exec
%%Pages: 1
(%%[ LastPage ]%%)  =
%%EOF
```

Exemple 2/3

- Le langage RTF

{\rtf1\adeflang1025\ansi\ansicpg1252\uc1\adeff0\deff0\stshfdbch0\stshfloch0\stshfhich0
\stshfbi0\deflang1036\deflangfe1036{\fonttbl{\f0\froman\fcharset0\fprq2{*\panose
020206030504050203 04}Times New Roman;}{\f36\froman\fcharset238\fprq2 Times
New Roman CE;}{\f37\froman\fcharset204\fprq2 Times New Roman Cyr;}…
{\f43\froman\fcharset186\fprq2 Times New Roman Baltic;}
{\f4
{\cc
en2
{\st
…;}
{*\
{*\
{\op
\pap
\def
…
{\rt
{\rt

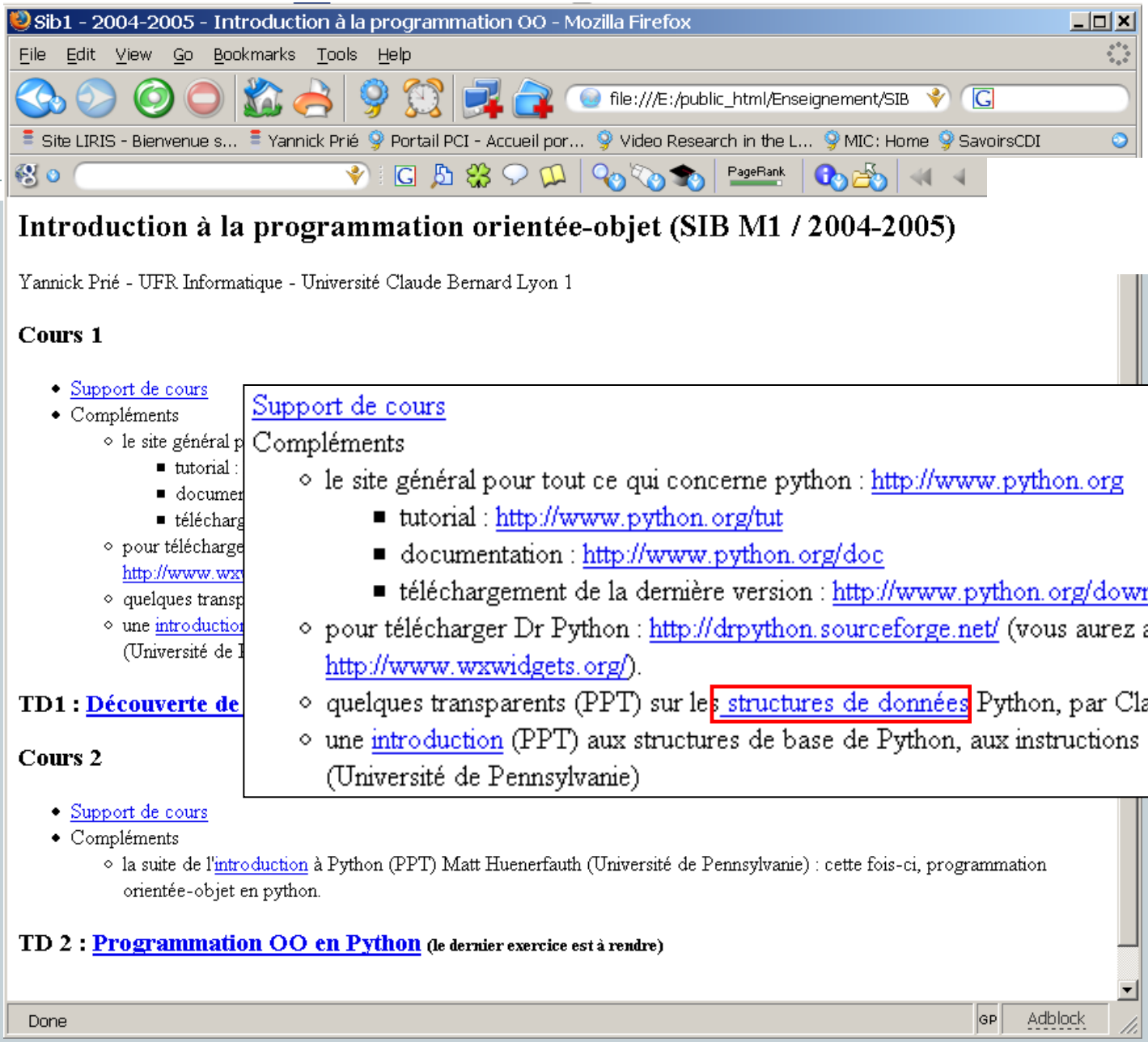


Exemple 3/3

- HyperText Markup Language (HTML)

```
    <ul>
      <li>tutorial : <a href="http://www.python.org/tut">http://www.python.org/tut</a></li>
      <li>documentation : <a href="http://www.python.org/doc">http://www.python.org/doc</a></li>
      <li>t&eacute;l&eacute;chargement de la derni&egrave;re version :
            <a href="http://www.python.org/download">http://www.python.org/download</a><br>
      </li>
    </ul>
  </li>
  <li>pour t&eacute;l&eacute;charger Dr Python :
      <a href="http://drpython.sourceforge.net/">http://drpython.sourceforge.net/</a>
    (vous aurez aussi besoin de la librairie graphique WxWidget :
          <a href="http://www.wxwidgets.org/">http://www.wxwidgets.org/</a>).</li>
  <li>quelques transparents (PPT) sur les<a href="http://cmsdoc.cern.ch/%7Egrandic/pythonintro.ppt">
    structures de donn&eacute;es</a> Python, par Claudio Grandi (Universit&eacute;
    de Bologne)</li>
  <li>une <a href="http://www.cis.upenn.edu/~cse391/cse391_2004/PythonIntro1.ppt">introduction</a>
    (PPT) aux structures de base de Python, aux instructions de base, et &agrave;
    la syntaxe, par Matt Huenerfauth (Universit&eacute; de Pennsylvanie)</li>
</ul>
```

lien


Sib1 - 2004-2005 - Introduction à la programmation OO - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///E:/public_html/Enseignement/SIB
Site LIRIS - Bienvenue s... Yannick Prié Portail PCI - Accueil por... Video Research in the L... MIC: Home SavoirsCDI
PageRank
Introduction à la programmation orientée-objet (SIB M1 / 2004-2005)
Yannick Prié - UFR Informatique - Université Claude Bernard Lyon 1
Cours 1
Support de cours
Compléments
le site général pour tout ce qui concerne python : http://www.python.org
tutorial : http://www.python.org/tut
documentation : http://www.python.org/doc
téléchargement de la dernière version : http://www.python.org/download
pour télécharger Dr Python : http://drpython.sourceforge.net/ (vous aurez aussi bes
http://www.wxwidgets.org/).
quelques transparents (PPT) sur les structures de données Python, par Claudio Gr
une introduction (PPT) aux structures de base de Python, aux instructions de base
(Université de Pennsylvanie)
TD1 : Découverte de
Cours 2
Support de cours
Compléments
la suite de l'introduction à Python (PPT) Matt Huenerfauth (Université de Pennsylvanie) : cette fois-ci, programmation orientée-objet en python.
TD 2 : Programmation OO en Python (le dernier exercice est à rendre)
Done
GP
Adblock

Types de langages à balises

- Langages de *balisage descriptif*
  - se contentent de décrire les données, sans but de traitement
  - ex. : RDF, OWL, MathML, n'importe quelle structure documentaire

Exemple

- Décrire une notice bibliographique
  - notice, titre, auteur, mots-clés, terme, résumé, ...
- Décrire un poème :
  - poeme, quatrain, tercet, vers, ...

```
<notice>
<auteur nat="fr">Victor Hugo
</auteur>
<titre>Quatre vingt treize</titre>
<mots-clés>
<terme>navire</terme>
<terme Type="pers">Marat</terme>
<terme Type="lieu">Paris</terme>
</mots-clés>
</notice>
```

```
<poeme type="sonnet">
<quatrain>
<vers>Je vis, je meurs ; je me brûle
et me noie.</vers>
<vers>J'ai chaud extrême en
endurant froidure ; </vers>
<vers> ... </vers>
<vers> ... </vers>
</quatrain>
...
</poeme>
```

→ vocabulaires différents

→ grammaires différentes

→ mais *même manière d'exprimer les descriptions*

Types de langages à balises

- *Méta-langages* de balisage
  - Langage avec lequel on peut définir d'autres langages
  - Pour les langages à balises
    - langage exprimant la manière dont on peut décrire une famille de langages à balise
      - comment exprimer les éléments ?
      - comment organiser les éléments ?
  - Exemples de métalangages
    - SGML
      - permet de définir : TEI, HTML…
    - XML
      - permet de définir : SVG, TEI, XHTML…

Plan

- Préambule
- Introduction
- Balise et balisage
- **Structure et arborescence**
- Standard Generalized Markup Language (SGML)
- Extensible Markup Language (XML)
- Conclusion

Structuration par balisage

- Idée générale
  - La structure est « ajoutée » au texte à l'aide de balises

```
<p>Il est de tradition de présenter un langage de programmation à l'aide d'un premier exemple comme : <eg> CHAR*20 GRTG GRTG = 'BONJOUR TOUT LE MONDE' PRINT *, GRTG END </eg></p> <p>Dans cet exemple, on commence par déclarer la variable <ident>GRTG</ident>, dans la ligne <kw>CHAR*20 GRTG</kw>, qui identifie <ident>GRTG</ident> comme formée de 20 octets de type <kw>CHAR</kw>. On affecte alors à cette variable la valeur <mentioned>BONJOUR TOUT LE MONDE</mentioned>. Suivent alors l'ordre d'impression <kw>PRINT</kw> et l'instruction finale <kw>END</kw>.</p>
```

p : servira à la mise en page

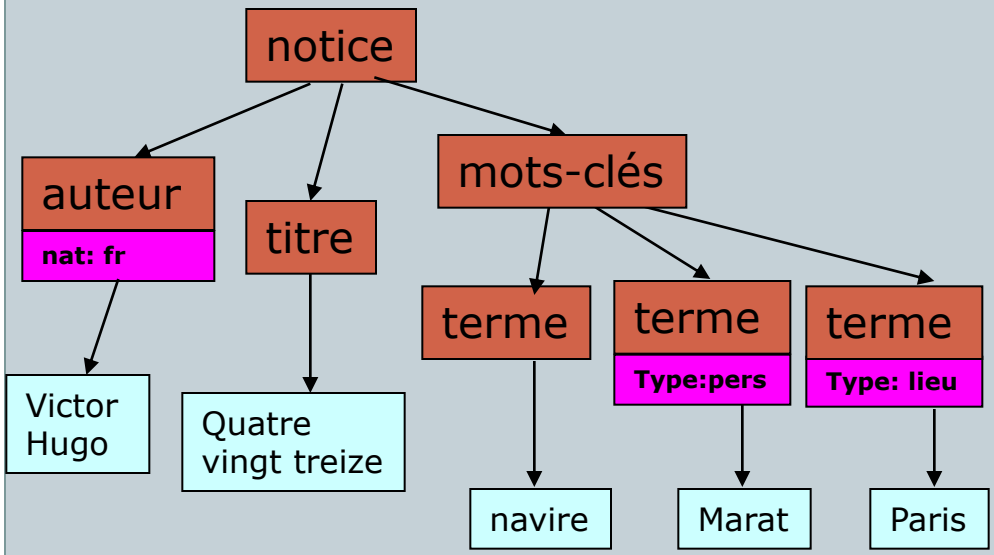
eg, kw,mentioned : seront mis en évidence dans la structure physique

kw,mentioned : utilisés pour construire un index

etc.

Notion d'arbre

- Principe
  - Représenter de l'information dans des structures arborescentes
  - Coder ces structures dans des fichiers, qui pourront être échangés



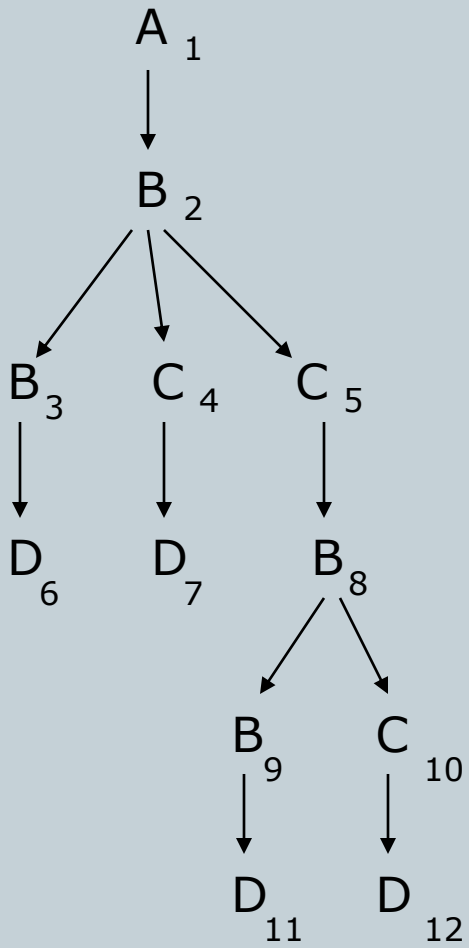

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<notice>
<auteur nat="fr">Victor Hugo
</auteur>
<titre>Quatre vingt treize</titre>
<mots-clés>
<terme>navire</terme>
<terme Type="pers">Marat</terme>
<terme Type="lieu">Paris</terme>
</mots-clés>
</notice>
```

Notion d'élément

- Problème
  - Un fichier est une suite d'octets

  ➔Comment représenter une structure d'arbre dans un fichier ?
- Solution
  - décrire l'arbre comme un ensemble d'éléments qui se contiennent les uns les autres.
  - représenter les éléments entre deux balises
    - balises ouvrantes
      - on les notera par exemple `<nom>`
    - balises fermantes
      - on les notera par exemple `</nom>`

Exemple



Eléments

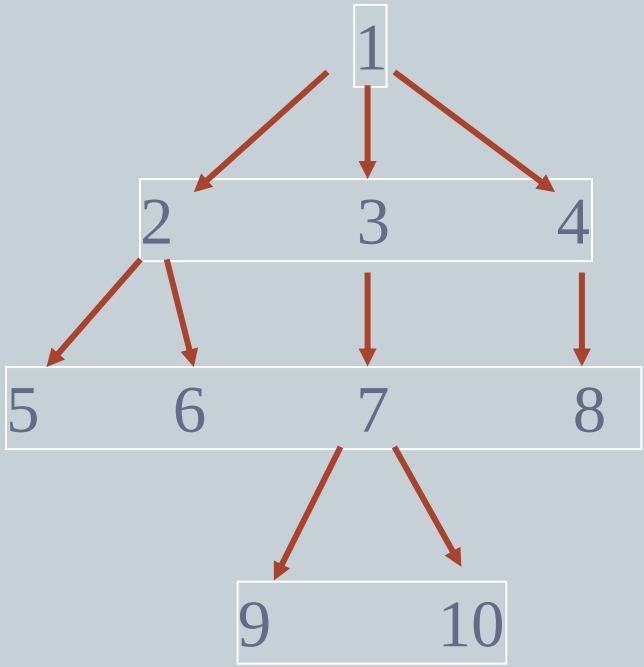
$A_1 \subset B_2$
$B_2 \subset B_3\ C_4\ C_5$
$B_3 \subset D_6$
$C_4 \subset D_7$
$C_5 \subset B_8$
$B_8 \subset B_9\ C_{10}$
$B_9 \subset D_{11}$
$C_{10} \subset D_{12}$

Eléments et balises

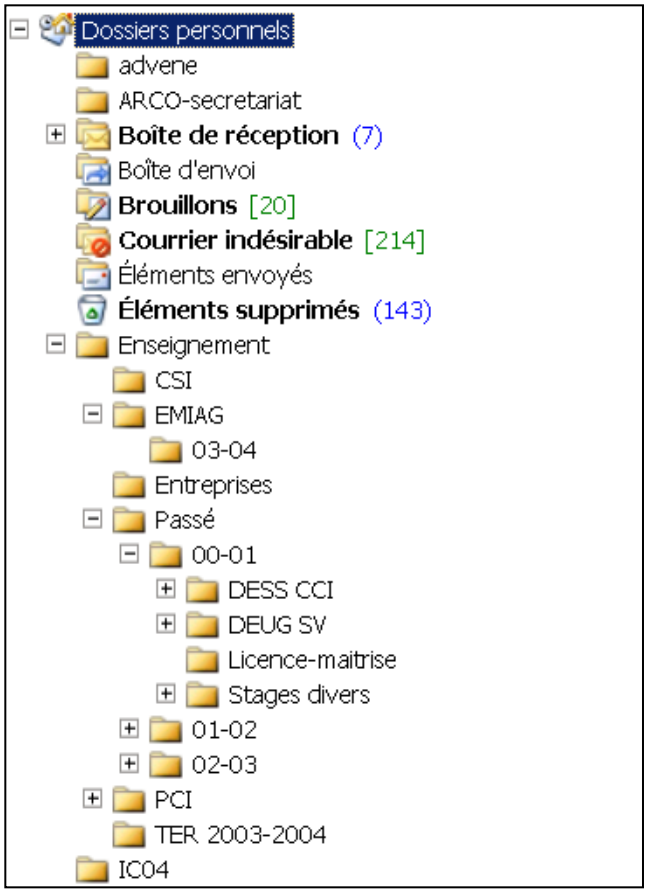
```
<A>
  <B>
    <B>
      <D></D>
    </B>
    <C>
      <D></D>
    </C>
    <C>
      <B>
        <B><D></D></B>
        <C><D></D></C>
      </B>
    </C>
  </B>
</A>
```

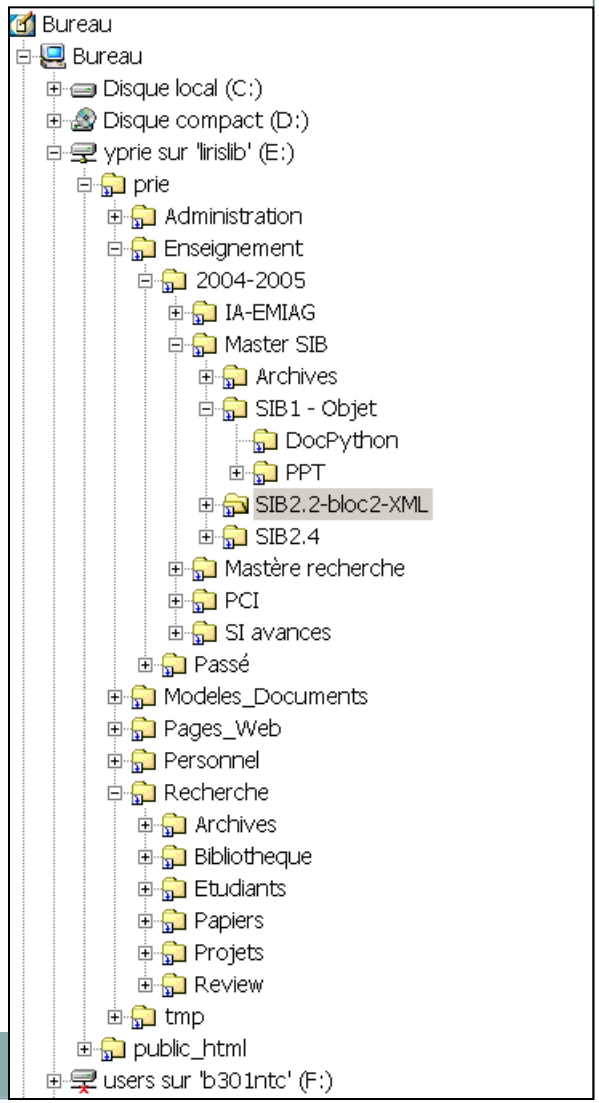
Parler des arbres

- Arbre
- Noeud
  - nœuds fils et pères
- Racine
- Feuille
- Chemin
  - suite de nœud
- Branche
  - chemin se terminant sur une feuille
- Ancêtres et descendants
- Taille d'un arbre
  - nombre de nœuds
- Profondeur d'un nœud



Les arbres sont partout !





chemin
E:\prie\Enseignement\2004-2005\
Master SIB\SIB3.2-bloc2-XML

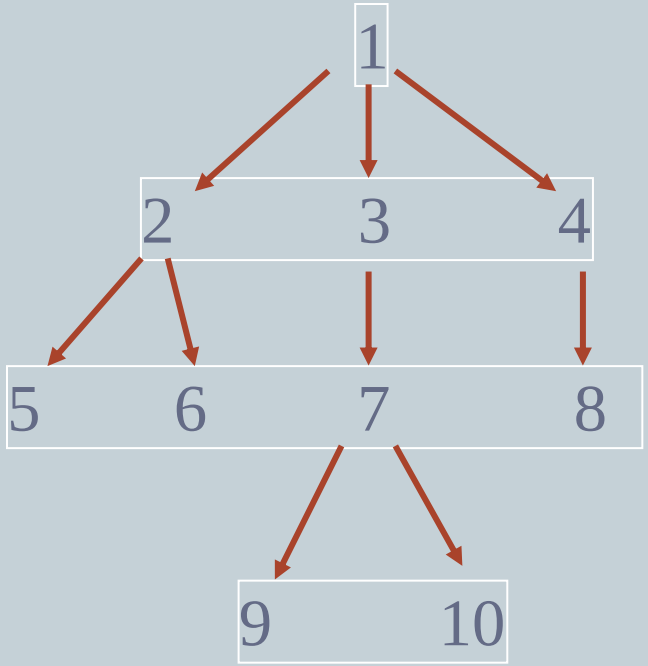
Parcours d'arbre

- Largeur d'abord

  1
  → 2 → 3 → 4
  → 5 → 6 → 7 → 8
  → 9 → 10

- Profondeur d'abord

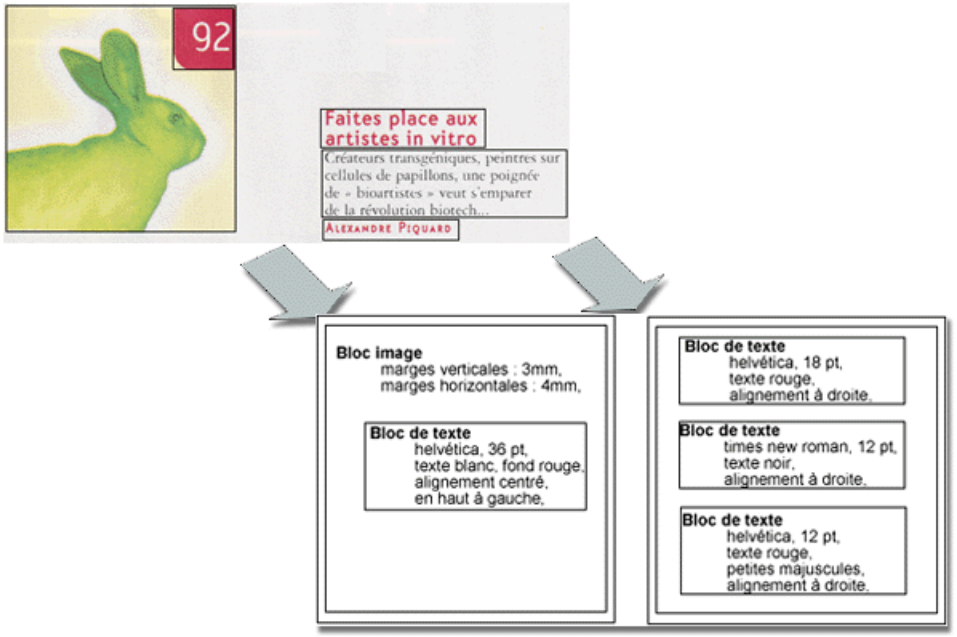
  1 → 2 → 5 → 6
  → 3 → 7 → 9 → 10
  → 4 → 8

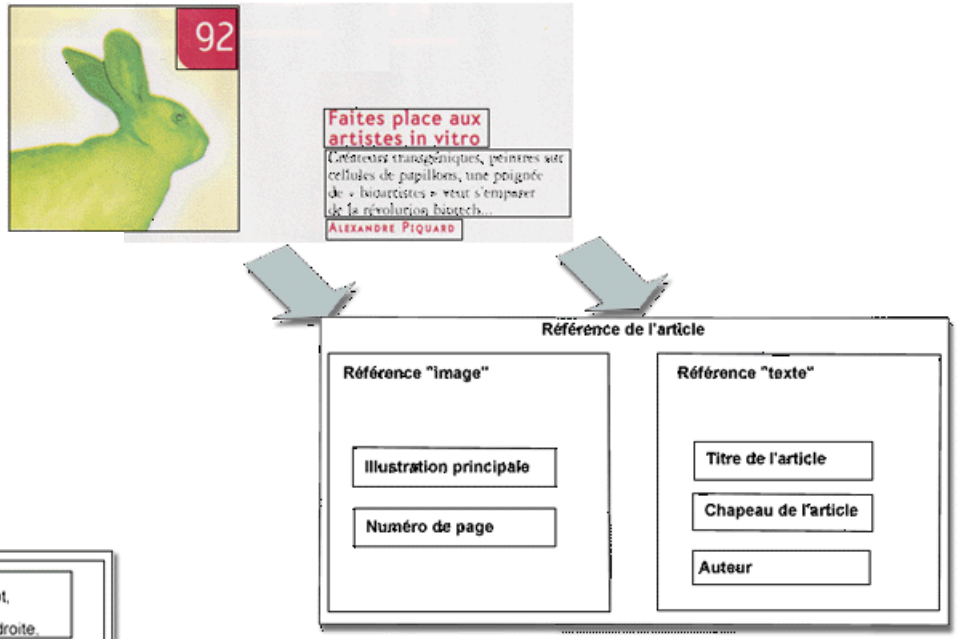


Différents types de structures

- Dichotomies
  - Structure physique / structure logique
  - Forme / contenu

Structure physique :
éléments de présentation





Structure logique :
organisation des contenus

(d'après http://sophia.univ-lyon2.fr/didacticiel/unite1/module2.html)

Plan

- Préambule
- Introduction
- Balise et balisage
- Structure et arborescence
- **Standard Generalized Markup Language (SGML)**
- Extensible Markup Language (XML)
- Conclusion

Remarque sur la normalisation

- Norme industrielle
  - Référentiel publié par un organisme officiel (ISO, AFNOR…).
  - En anglais : *standard*
- Standard
  - Référentiel publié par une entité privée
  - Si diffusion large : standard *de fait*
- Consortium
  - Ensemble d'entreprises, de centres de recherche, de particuliers qui s'allient pour définir des normes et standards sur tout et n'importe quoi
  - Gain : fournir les outils au moment où le référentiel est publié
    - JPEG (Joint Picture Expert Group) → norme ISO
    - MPEG (Moving Picture Expert Group) → norme ISO
    - W3C (World Wide Web Consortium) → standards
    - …

SGML

- Objectif : représenter l'information contenue dans un document indépendamment
  - des systèmes utilisés pour la saisie et le traitement
  - de la forme physique qu'il sera amené à prendre (papier, CD-ROM, web…)
  - des langues et des alphabets, latins ou non
  - des applications
- Naissance chez IBM (années soixante)
  - GML
  - gestion de la documentation technique
- Normalisation 1986 ISO-8879
  - une dizaine d'années de travail
- Utilisation
  - Description des documents dans les grosses organisations
    - complexité des langages
    - lourdeur et cherté des outils (chaîne de traitement)
    - Journal Officiel, grosses entreprises/documentations, éditeurs…
  - Échange des documents

SGML : principes

- Métalangage
  - permet de décrire des modèles (grammaires)
- Notion de DTD
  - Document Type Definition
  - Permet de décrire un modèle
    - un type de document
- Un document SGML
  - Est une instance du type de document
  - Doit être conforme à la DTD associée

SGML : exemple

Instance

```
<!DOCTYPE memo SYSTEM "memo.dtd">
<memo statut="conf">
<auteur>Serge Fleury</auteur>
<dest>
<nom>André Salem</nom>
<nom>Pollet Samvelian</nom>
</dest>
<sujet>Cours SLFE6</sujet>
<corps>
<par>Veuillez noter que le cours SLFE6 sur
    les documents électronique aura bel et
    bien lieu au mois de mai 2002</par>
<par>S'il y avait des changements de votre
    côté, veuillez m'en aviser dans les plus
    brefs délais.</par>
</corps>
</memo>
```

DTD (memo.dtd)

```
<!-- DTD utilisable pour baliser les memos en
    SGML -->
<!ELEMENT memo -- ((auteur & (date?) &
    sujet & dest & (cc?)), corps)>
<!ATTLIST memo statut (conf | pub) pub>
<!ELEMENT (dest | cc) -- (nom+)>
<!ELEMENT corps -- (par*)>
<!ELEMENT (auteur | date | sujet | nom |
    par) -- (#PCDATA)>
```

Un élément corps contient un nombre quelconque de paragraphes

Un élément dest ou cc contient au moins un nom

(d'après http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/)

Plan

- Préambule
- Introduction
- Balise et balisage
- Structure et arborescence
- Standard Generalized Markup Language (SGML)
- **Extensible Markup Language (XML)**
  - **Généralités**
  - Syntaxe et documents bien formés
  - Construire un document XML
  - Traiter automatiquement des documents XML
- Conclusion

Extensible Markup Language (XML)

- Position du problème
  - représenter et échanger des données et des documents sur le web
- SGML
  - un peu vieux
  - trop complexe
- HTML
  - trop basique
    - document = en-tête + corps
  - mélange logique / présentation
    - balise b = bold (mise en gras) :

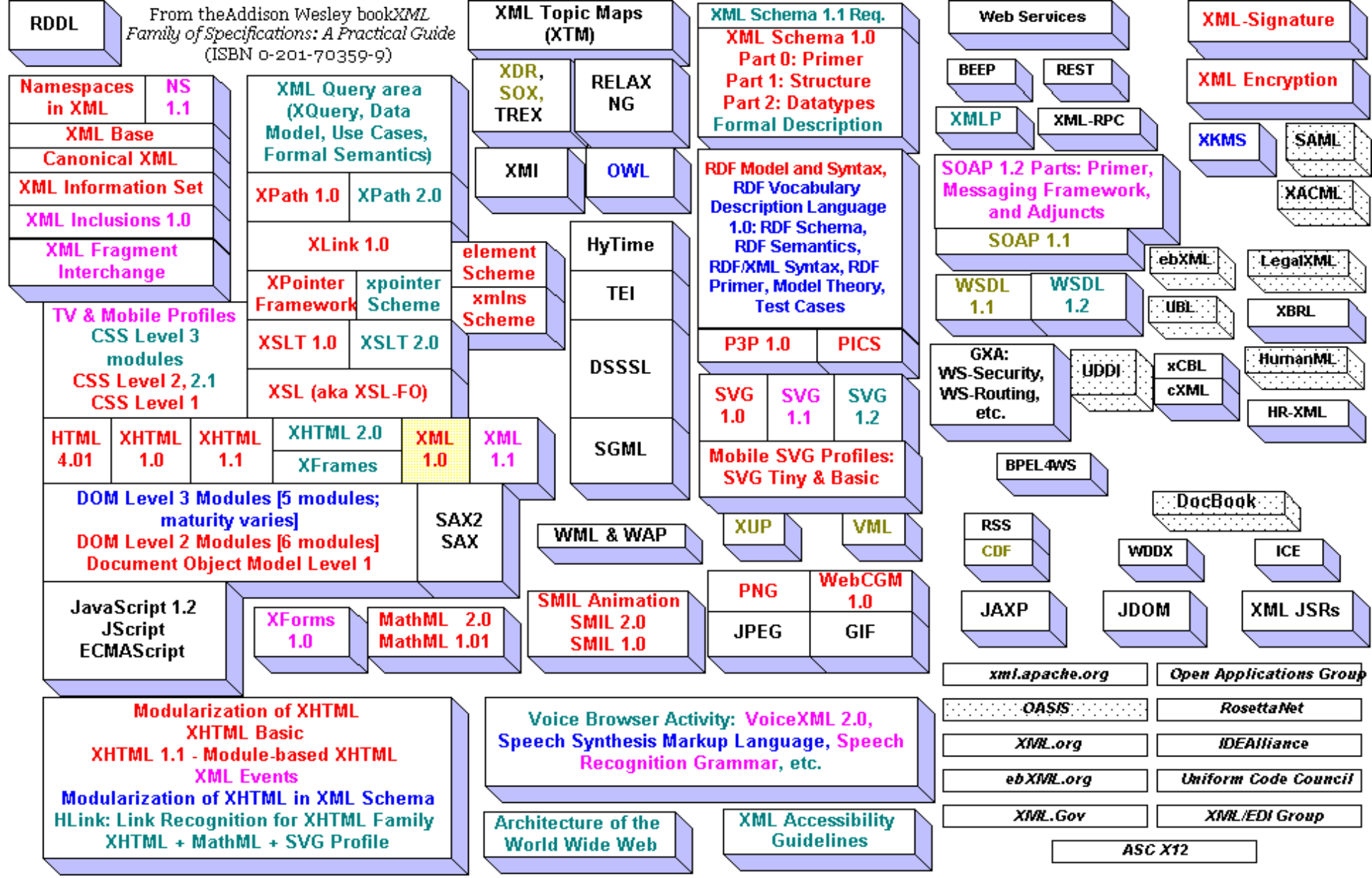
      Attention !
    - bonne approche
      - <important>Attention !</important>
      - présenter la chaîne de caractères importante avec une mise en forme particulière (italique, rouge, gras, *etc.*)

XML : prérequis

- XML doit être facilement utilisable sur le Web
- XML doit supporter une grande variété d'applications
- XML doit être compatible avec SGML
- Il doit être facile d'écrire des programmes qui traitent des documents XML
- Le nombre d'options doit être réduit au minimum, idéalement à zéro
- Les documents XML doivent être lisibles et raisonnablement clairs
- La conception de XML doit être menée rapidement
- La description de XML doit être formelle et concise
- Les documents XML doivent être faciles à créer
- La concision du balisage XML est d'une importance minime

XML : principes de base

- DTD de SGML (beaucoup plus concis)
  - Restrictions de syntaxe et non de contenu
- Versions
  - V 1.0 : fév. 1998 -> 04 fév. 2004 (3ème édition)
  - V 1.1 : 04 fév. 2004 ; 16 août 2006 (2ème éd.)
- Au même titre que SGML, c'est un méta-langage de description des données
  - ⇒ Ça ne sert à rien
  - ⇒ Ça permet de définir des « applications » (types de documents) pour faire ce qu'on veut avec
    - Visualiser des pages web
    - Décrire des images…
  - Chaque document correspondant à une application est appelé « instance »



The XML Family of Specifications: The Big Picture

Last Updated: April 19, 2003

Recommendation	Proposed Recommendation	Candidate Recommendation	Last Call Working Draft	Working Draft	Note submitted to W3C	Not a W3C specification

Plan

- Préambule
- Introduction
- Balise et balisage
- Structure et arborescence
- Standard Generalized Markup Language (SGML)
- **Extensible Markup Language (XML)**
  - Généralités
  - **Syntaxe et documents bien formés**
  - Construire un document XML
  - Traiter automatiquement des documents XML
- Conclusion

Un document XML


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE livre SYSTEM "D:\Enseignement\2004-
  2005\SIB3.2\CM\exemple-intro.dtd">
<livre id="561" nbpages="190" titre="La compagnie des spectres">
  <auteur>
       <nom>Salvayre</nom>
       <prenom>Lydie</prenom>
  </auteur>
  <format type="poche">
       <mesure type="largeur" unite="cm">11</mesure>
       <mesure type="longeur" unite="cm">19</mesure>
       <mesure type="hauteur" unite="mm">10</mesure>
  </format>
</livre>
```

La DTD correspondante


```
<!ELEMENT livre (auteur, format)>
<!ATTLIST livre
  id CDATA #REQUIRED
  nbpages CDATA #REQUIRED
  titre CDATA #REQUIRED >
<!ELEMENT auteur (nom, prenom)>
<!ELEMENT format (mesure+)>
<!ATTLIST format
  type CDATA #REQUIRED >
<!ELEMENT mesure (#PCDATA)>
<!ATTLIST mesure
  type (hauteur | largeur | longeur) #REQUIRED
  unite (cm | mm | in) #REQUIRED >
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

Qu'y a-t-il dans un document XML ?

- Prologue
  - Déclaration XML
  - Déclarations de DTD
    - Instructions pour les processeurs XML
  - Instructions de traitement
    - Instructions pour applications externes
- Arbre des éléments
  - Eléments
    - Balises XML pour le marquage
    - Contenu
      - texte
      - autres éléments
  - Attributs des éléments
    - Information associées aux éléments
- Commentaires

Déclaration XML

- Syntaxe générale

```
<?xml version="1.0" [encoding = "encodage"] [standalone="yes|no »] ?>
```

- Est une des informations de traitement
- Indique
  - Conformité du document à une version de la norme XML
    - version="1.0"
  - Jeu de caractères utilisé dans le document
    - encoding = "UTF-8"
  - Présence ou non de références externes
    - standalone="yes"

Instructions de traitement

- Informations nécessaires à une application externe
- Format :
  - `<?NomApplication paramètres ?>`
- Exemples
  - Déclaration de feuille de style à utiliser

    ```
    <?xml-stylesheet href="fichier.xsl" type="text/xsl"?>
    ```
  - Déclaration XML de début de fichier

    ```
    <?xml version='1.0' ?>
    ```

Eléments : règles de base

- Un nom d'élément
  - commence par une lettre ou souligné
  - contient des lettres, chiffres, et "-", ".", ":", "_"
  - peut posséder un préfixe
    - `prefixe:nom_element`
    - Ex. : `xsl:template`
- Les noms d'éléments dépendent de la casse
  - `<nom_element>` ≠ `<nom_Element>`
- Balises
  - de début : `<nom_element>`
  - de fin : `</nom_element>`
- Les éléments peuvent être vides
  - pas de contenu
  - `<element_vide />`
  - Ex : `<img src= "toto.jpg" />`

Arbre des éléments

- Un seul élément racine qui contient tous les autres
- Pas d'intersections entre éléments
  - Mauvais : `<nom1><nom2>…</nom1></nom2>`
  - Bon : `<nom1><nom2>…</nom2></nom1>`
- Blancs ou retours chariot en général non significatifs
  - `<section><p> … </p></section>`
  - ```
    <section>
       … 
    </section>
    ```
- Les éléments sont ordonnés
    ```

Caractères spéciaux

- Ces caractères ont une signification spéciale pour les outils XML
- Il faut les écrire différemment
  - < <
  - > >
  - & &
  - ' '
  - " "

Attributs associés aux éléments : règles de base

- Dans les balises ouvrantes
  - `<el att1="valeur1" att2="valeur2">`
- Les noms d'attributs dépendent de la casse
  - `<el att1="valeur1" Att1="valeur2">`
- Valeurs d'attributs entourées
  - par des guillemets (") ou des apostrophes (')
- Les attributs sont non-ordonnés

Attributs

- Les valeurs peuvent être
  - des données textuelles
    - `value="N'importe quoi"`
  - des *tokens* (noms XML) simples
    - `value = "blue"`
  - des ensemble *de tokens*
    - `value = "red green blue"`
- Possibilité d'énumérer les valeurs possibles et de mettre des valeurs par défaut (voir DTD)

Attributs de type ID et IDREF(S)

- Permettent des relations non hiérarchiques entre éléments
  - ID : identificateur unique dans le document XML
  - IDREF : référence à un élément ayant un attribut de type ID
  - IDREFS : références à *des* éléments ayant un attribut de type ID
- Exemple

```
<société codes_services="A001 A003">
<service code="A001">
<employé code="E205" code_service="A001"> Jean Dupont </employé>
<employé code="E206" code_service="A001"> Frédéric Marc </employé>
<employé code="E207" code_service="A001"> Fabrice Detterre </employé>
<employé code="H107" code_service="A003"> Angélique Millet </employé>
</service>
<service code="A003">
<employé code="A115" code_service="A003"> Isabelle Mascot </employé>
</service>
</société>
```

- Exercice
  - Construire l'arbre correspondant, dessiner les relations entre référence vers des identificateurs (idrefs) et identificateurs (id)

Commentaires

- Les commentaires ne sont pas considérés comme faisant partie du document XML
  - `<!-- Un commentaire -->`
- Pas de '--' dans un commentaire !
- Un commentaire ne peut pas se trouver dans une autre déclaration

Au bilan : un document XML bien formé

- Prologue
  - en-tête
  - déclaration de DTD *(pas encore vu)*
  - instructions de traitement
- Eléments
  - attributs
  - contenus
- Commentaires

Plan

- Préambule
- Introduction
- Balise et balisage
- Structure et arborescence
- Standard Generalized Markup Language (SGML)
- **Extensible Markup Language (XML)**
  - Généralités
  - Syntaxe et documents bien formés
  - **Construire un document XML**
  - Traiter automatiquement des documents XML
- Conclusion

Identifier les données qui nécessitent d'être balisées

- Pour chaque unité d'information, déterminer
  - Peut-on lui donner un nom ?
  - Apparaît-elle tout le temps ?
  - Peut-il y en avoir plusieurs ?
  - Peut-on la décomposer en des unités plus petites ?
  - Y-a-t'il du contenu textuel qui ne change pas ?
  - Comment est-elle associée aux autres unités ?

Granularité

- <PERSON>
  <NAME>Jon Smith</NAME>
  </PERSON>

- <PERSON>
  <FORENAME>Jon</FORENAME>
  <SURNAME>Smith</SURNAME>
  </PERSON>

Eléments ou attributs ?

- Comment les données doivent-elles être encapsulées ?
  - ```
    <book>
      <title>The Forty-nine Steps</title>
      …
    </book>
    ```
  - ```
    <book title="The Forty-nine Steps">
      …
    </book>
    ```
- Tout dépend de ce que l'on veut faire…
- Il existe des avis tranchés…

Eléments ou attributs ? (2)

- Séparer le contenu des métadonnées
  - Données qui doivent être imprimées comme du texte comme contenu
  - Métadonnées comme attributs
- Règles générales
  - Si on enlève toutes les balises, le document doit encore être lisible et utilisable
  - S'il y a doute, utiliser un attribut

Plan

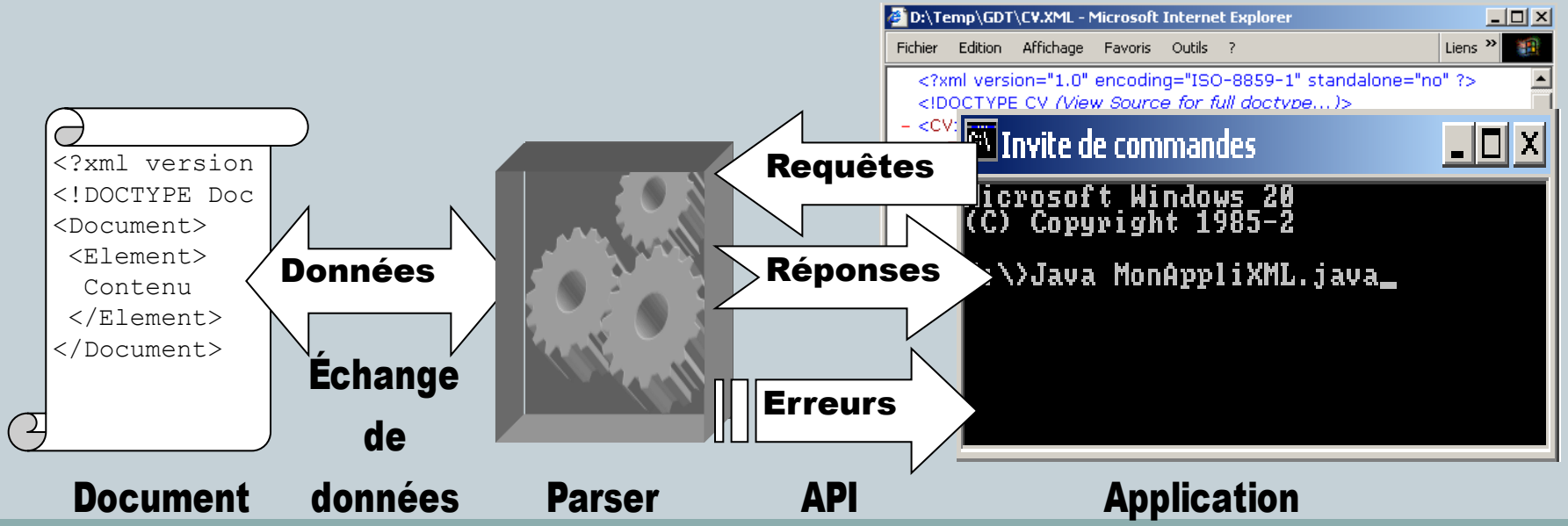
- Préambule
- Introduction
- Balise et balisage
- Structure et arborescence
- Standard Generalized Markup Language (SGML)
- **Extensible Markup Language (XML)**
  - Généralités
  - Syntaxe et documents bien formés
  - Construire un document XML
  - **Traiter automatiquement des documents XML**
- Conclusion

Notion de parseur XML

- Outil qui lit un document XML et le met à disposition d'une autre application
  - Anglicisme d'après *parser* (analyseur)
  - Aussi appelé « processeur »
- 2 rôles distincts
  - Vérifier qu'un document répond bien à la syntaxe XML
    - Document *bien formé*
    - Possibilité de l'utiliser en tant que tel
      - ex. : le présenter à l'utilisateur
  - Vérifier qu'un document suit bien la grammaire définie (DTD / schéma)
    - Document *valide*
- Exemple de fonctionnement interne
  - Construit l'arbre des éléments et permet la navigation dans cet arbre

Outils de programmation avec XML

- Communications entre parsers et applications
  - Rappel : Application Programming Interface
    - Outils
    - Protocole de communication
  - Schéma des échanges de données

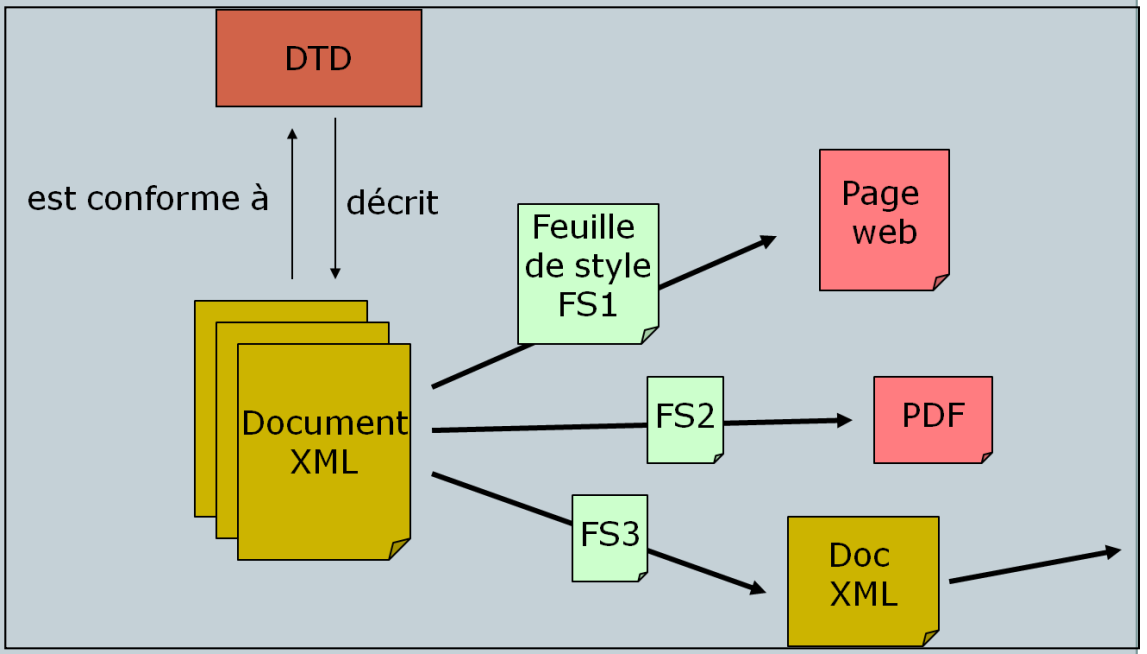


Plan

- Préambule
- Introduction
- Balise et balisage
- Structure et arborescence
- Standard Generalized Markup Language (SGML)
- Extensible Markup Language (XML)
- **Conclusion**

Utilisation de XML : principe général

- DTDs, Schéma
  - comment décrire les données et les documents ?
- Documents XML
  - les données et les documents, eux-mêmes dans des fichiers
- Feuilles de style
  - manière de présenter les données et les documents
- Remarque
  - on ne sait plus bien où sont les données et où sont les documents



XML : utilisation dans les SI

- Stockage de données
  - simples fichiers (ex. configuration)
  - bases de données semi-structurées (requêtes, etc.)
  - bases de données documentaires
    - documents XML
    - documents XHTML (web)
- Echange de données
  - d'une base de données vers une autre (format d'échange)
  - serveur vers un navigateur : données + feuille de style
- Remarque :
  - circulation de flux XML sur un réseau :
    - utilisation de l'arbre entier (le document)
    - utilisation à la volée pour les très gros documents (exemple : BiM)

Conclusion

- Notions d'arbre et méta-langage XML à la base de ce cours
- Utilité
  - pour le Web, l'échange de données, le tagging de documents…
- Suite du module
  - XML : 2 CM / 3 TP
  - (X)HTML / CSS : 2 CM / 2 TP
  - XPATH, XSL : 2 CM / 2 TP