

Documents valides : Définition de type de document



M1 PANIST – 2011-2012
LIONEL MÉDINI
COURS 2

Université Claude Bernard Lyon 1
UFR FST – département Informatique

D'après le cours de Yannick Prié

Objectifs de ce cours

2

- Être capable de
 - Comprendre les DTD
 - Construire des documents XML et des DTD

Plan

3

- Documents XML
 - Syntaxe XML et documents bien formés
- **Types de documents XML**
 - **DTD et documents valides**
 - Introduction à XML-Schema
- Le monde XML
 - Quelques normes liées à XML
 - Quelques DTD importantes

Notion de grammaire

4

- **Définition**

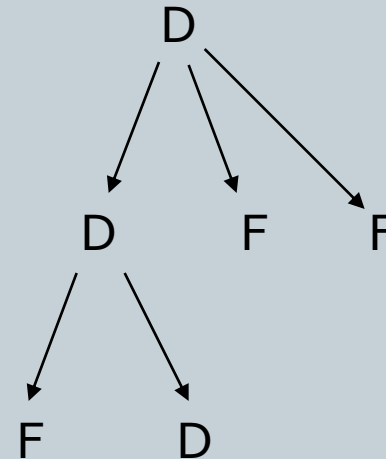
- Une grammaire est un système formel qui définit complètement un arbre

- **Contenu**

- vocabulaire
- règles de production
 - ✦ Avec une syntaxe spécifique

- **Exemple du système de fichiers**

- **Vocabulaire**
 - ✦ D (Dossier)
 - ✦ F (fichier)
- **Règles**
 - ✦ Départ : D
 - ✦ $D \rightarrow (D|F)^*$
- **Syntaxe des règles**

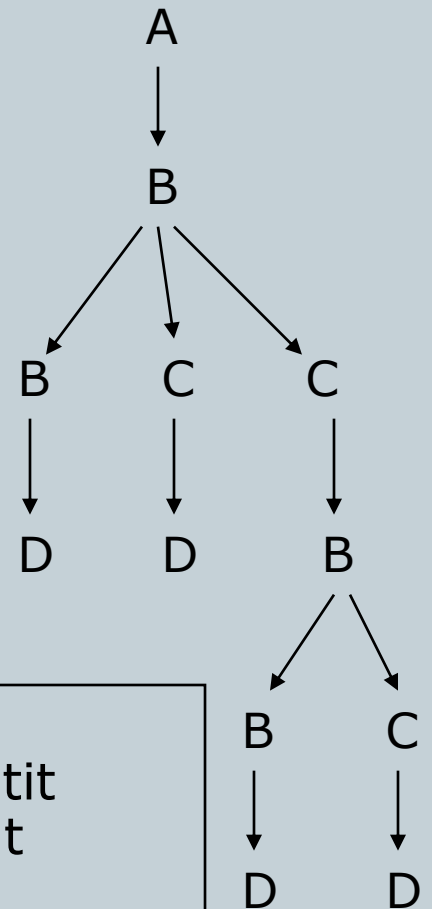


Règle	Signification
*	Zéro, un, plusieurs
	OU

Grammaire : autre exemple

5

- Vocabulaire
 - A, B, C, D
- Règles
 - Départ : A
 - $A \rightarrow B^+$
 - $B \rightarrow (B, C^*) \mid D$
 - $C \rightarrow (D \mid B)$
- Syntaxe des règles



Règle	Signification
*	Zéro, un, plusieurs
+	Un ou plusieurs
	OU
,	ET

→ Question
quel est le plus petit
arbre que l'on peut
écrire avec cette
grammaire ?

Document Type Definition

6

- Pour définir le type de document XML voulu
 - grammaire pour décrire comment construire un document XML
- Permet de
 - valider un document XML (avec un parseur validant)
 - ✦ vérifier que tous les éléments sont présents et corrects
 - ✦ vérifier que les noms d'attributs et leurs valeurs sont corrects
 - transmettre cette connaissance à d'autres
 - ✦ ils pourront définir leurs propres documents XML suivant la même DTD
 - ✦ d'où possibilité de standardisation et d'échanges

Exemple : un document XML

7

```
<!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 | longueur) #REQUIRED
```

```
  unite (cm | mm | in) #REQUIRED >
```

```
<!ELEMENT nom (#PCDATA)>
```

```
<!ELEMENT prenom (#PCDATA)>
```

← Sa DTD

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE livre SYSTEM "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="longueur" unite="cm">19</mesure>
```

```
    <mesure type="hauteur" unite="mm">10</mesure>
```

```
  </format>
```

```
</livre>
```

Un document XML →

DTD

8

- **Déclaration**
 - contenant la définition formelle de la structure autorisée,
 - décrit
 - ✦ quels noms sont utilisés pour les types d'éléments
 - ✦ comment ces types d'éléments s'organisent
 - ordre
 - hiérarchie
 - ✦ les attributs des éléments
 - ✦ des entités analysables ou non
 - ✦ des notations pour les types de données binaires
- **Liaison DTD / document XML**
 - soit la DTD est dans le document XML (inline)
 - soit le document XML réfère à la DTD avec une URL (la DTD est dans un fichier externe)

Langage de DTD

9

- Un autre langage que XML
 - autre syntaxe, mots-clés différents
 - origines dans SGML
- Pour décrire des types de documents XML
- Remarque
 - XML schema permet de faire la même chose, en XML

Déclarations

10

- Instructions pour le processeur XML
- Format : `<! ... >` ou `<! ... [<! ... >]>`
 - Document type - `<!DOCTYPE ... >`
 - Character data - `<![CDATA[...]]>`
 - Entities - `<!ENTITY ... >`
 - Notation - `<!NOTATION ... >`
 - Element - `<!ELEMENT ... >`
 - Attributes - `<!ATTLIST ... >`
 - `<![INCLUDE [...]]>` et `<![IGNORE [...]]>`

Déclaration *Document Type*

11

- Placée au début du document XML
- Identifie le nom de *l'élément racine* du document
- Permet aussi de rajouter des définitions d'entités et des DTD

- `<!DOCTYPE Racine>`

- `<!DOCTYPE Racine [...] >`

- `<Racine>`

- `...`

- `</Racine>`

←
déclaration
de la DTD

Déclaration *Character Data*

12

- Placée dans le document XML
- Permet de taper directement du texte qui ne doit pas être interprété par un outil XML
- Deux textes équivalents
 - `Press <<<ENTER>>>`
 - `<![CDATA[Press <<<ENTER>>>]]>`

DTD et document XML

13

Valide,
contraint



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE livre SYSTEM "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="longueur" unite="cm">19</mesure>
    <mesure type="hauteur" unite="mm">10</mesure>
  </format>
</livre>
```

Document XML

```
<!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 |
longueur) #REQUIRED
  unite (cm | mm | in)
#REQUIRED >
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

DTD



Est validé par,
Est contraint par

Mettre en place une DTD

14

- Définir différents composants XML...
 - Entités, éléments, déclarations, instructions de traitements, listes d'attributs, *etc.*
- ... dans des DTD pour spécifier les règles permettant de valider des documents XML
 - Définir un modèle (type) de document de façon formelle
 - ✦ pour qu'un parser puisse valider les documents qui lui seront soumis
- Une DTD décrit
 - Quels noms peuvent être utilisés pour les types d'éléments
 - L'ordre dans lesquels ceux-ci peuvent apparaître
 - La hiérarchie documentaire
 - Les noms et les types des attributs d'éléments

Déclaration de DTD

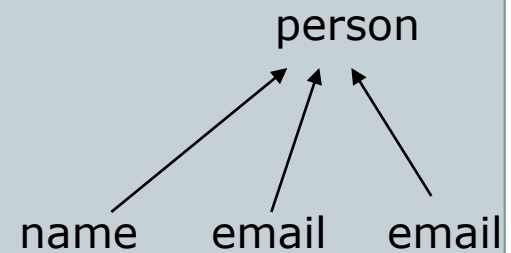
15

- **La DTD est stockée**
 - soit dans le fichier XML
 - soit dans un fichier extérieur
 - soit dans les deux
- **Une DTD interne peut écraser ou ajouter des ENTITY ou des ATTLIST à des définitions de DTD externes**
 - principe : “le dernier qui parle a raison”
- **Une DTD est composée de déclarations**
 - ELEMENT – définitions d’éléments
 - ATTLIST – définitions d’attributs
 - ENTITY – définitions d’entités
 - NOTATION – définitions de notations

Déclarations d'éléments

16

- Définir un élément et son contenu
 - `<!ELEMENT name (#PCDATA)>`
⇒ `<name> ...du texte... </name>`
- Définir un élément vide (sans contenu)
 - `<!ELEMENT name EMPTY>`
⇒ `<name/>`
- Définir des éléments fils spécifiques
 - `<!ELEMENT person (name, email*)>`
 - Éléments fils quelconques
 - ✦ `<!ELEMENT name ANY>`



Définir une hiérarchie (grammaire)

17

- Définir le contenu des éléments

```
<!ELEMENT person (name, email*)>
```

- ...et définir une hiérarchie d'éléments

```
<!ELEMENT name (fname, surname)>
```

```
<!ELEMENT fname (#PCDATA)>
```

```
<!ELEMENT surname (#PCDATA)>
```

```
<!ELEMENT e-mail (#PCDATA)>
```



- Organisation des sous-éléments

- Connecteur de séquence ',': (A, B, C) [puis]

- Connecteur de choix '|': (A | B | C) [ou]

Indicateurs de quantité (“cardinalité”)

18

- Quantité d’éléments fils d’un élément donné

A	Unique	[1 ...1]
A?	Possible	[0...1]
A+	Au moins 1 fois	[1 ...*]
A*	Peu importe...	[0...*]

- Exemples

$(A, B) +$
 $((A, B?) \mid C+) *$

Déclaration d'attributs

19

- Les attributs sont associés aux éléments qui les contiennent
- Chaque type d'élément peut posséder une liste d'attributs
- Déclarés dans une déclaration ATTLIST liée à l'élément
 - `<!ELEMENT nom_element ... >`
 - `<!ATTLIST nom_element ... >`
 - Il faut ensuite définir
 - ✦ le nom de l'attribut
 - ✦ le type de l'attribut
 - ✦ sa valeur par défaut

Noms et types d'attributs

20

- Noms d'attributs
 - `<!ATTLIST elem nom type par_defaut>`
 - `<!ATTLIST elem attribut_1 ...
attribut_2 ...
attribut_3 ... >`
- Dix types possibles d'attributs

CDATA

NMTOKEN

NMTOKENS

ENTITY

ENTITIES

ID

IDREF

IDREFS

NOTATION

name group

Types d'attributs (1)

21

- **CDATA**
 - Chaîne de caractères
 - `<!ATTLIST person name CDATA >`
 - `<person name = "Tom Jones">`
- **NMTOKEN**
 - Token unique
 - `<!ATTLIST mug color NMTOKEN >`
 - `<mug color="red">`
- **NMTOKENS**
 - Tokens multiples
 - `<!ATTLIST temp values NMTOKENS >`
 - `<temp values="12 15 34">`

Joue sur la manière dont le parseur interprète l'attribut :

- comme une chaîne de caractères quelconque
- Comme une chaîne représentant un item d'une liste
- comme un ensemble d'items séparés par des espaces

Types d'attributs (2)

22

- **ID**

- Identificateur unique pour l'élément

- `<!ATTLIST person id ID >`

- `<person id = "P09567">Toto</person>`

- `<person id = "P09567">Tutu</person>` ← Nonvalide

- **IDREF**

- Référence à un ID

- `<!ATTLIST person father IDREF >`

- `<person father="P09567">`

- **IDREFS**

- Référence à plusieurs ID

- `<!ATTLIST person children IDREFS >`

- `<person children="P09567 P09677">`

Types d'attributs (3)

23

- **Name group**
 - Liste restreinte de valeurs possibles
 - `<!ATTLIST point coord (X|Y|Z) >`
 - `<point coord="X">`
- **ENTITY (substitut pour autre chose)**
 - L'attribut est une référence d'entité
 - `<!ATTLIST person photo ENTITY >`
 - `<person photo="MyPic">`
- **ENTITIES**
 - Plusieurs références d'entités
 - `<!ATTLIST album photos ENTITIES >`
 - `<album photos="pic1 pic2">`

Types d'attributs (4)

24

- **NOTATION**

- Décrit des données non XML

- `<!NOTATION jpeg SYSTEM "JPEG Image" >`

- `<!NOTATION gif PUBLIC "+//ISBN 0-7923-9432-1::Graphic Notation//NOTATION CompuServer Graphic Interchange Format//EN">`

- `<!ATTLIST image format NOTATION (jpeg|gif) ...>`

- `<image format="gif">`

Valeurs d'attributs par défaut

25

- Quatre types

- **#REQUIRED** *doit* être spécifié
- **#IMPLIED** *peut* être spécifié
- **"default"** valeur par défaut si non spécifié
- **#FIXED** seule valeur autorisée

```
<ATTLIST element nom type par_defaut>
<!ATTLIST citoyen parents IDREFS #REQUIRED
id ID #IMPLIED
sex (m|f) "f"
adress CDATA #IMPLIED
nat CDATA #FIXED "Fr">
```


Déclaration interne de DTD

26

- Dans la déclaration DOCTYPE

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE racine [
  <!-- ici la DTD -->
  <! ... >
  <! ... >
]>
<!-- ici le reste du fichier XML -->
<racine>
...
</racine>
```

Pas besoin de fichier
supplémentaire



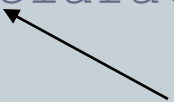
Déclaration de DTD externe privée

27

- Référence à la DTD externe par un chemin dans la déclaration DOCTYPE

- ```
<?xml version="1.0" standalone="no" ?> <!DOCTYPE
racine
 SYSTEM "dossiers/MyDoc.dtd" [
 <!-- déclarations supplémentaires -->
 <! ... >
 <! ... >
]>
<!-- ici le reste du fichier XML -->
```

DTD externe  
privée



- Les déclarations spécifiques au document restent définies de façon interne

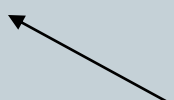
# Déclaration de DTD externe privée

28

- Référence à la DTD externe par une URL dans la déclaration DOCTYPE

- ```
<?xml version="1.0" standalone="no" ?> <!DOCTYPE
racine
  SYSTEM "http://serveur/chemin/MyDoc.dtd" [
  <!-- déclarations supplémentaires -->
  <! ... >
  <! ... >
]>
<!-- ici le reste du fichier XML -->
```

DTD externe
privée, URL



- Les déclarations spécifiques au document restent définies de façon interne

Déclaration de DTD externe publique

29

- Utilisation du mot-clé PUBLIC

```
<!DOCTYPE racine  
    PUBLIC "identifiant public" "url" >
```

- Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html>  
.....  
</html>
```

Identificateurs publics

30

```
PUBLIC "-//EBI//DTD My book//EN" "url"
```

- PUBLIC keyword
- Identifier type - Registered + / - / ISO
- Owner identifier
- Public text class
 - DTD, NOTATIONS, ENTITIES, TEXT
- Public text description
- Public text language
- url non obligatoire mais conseillée

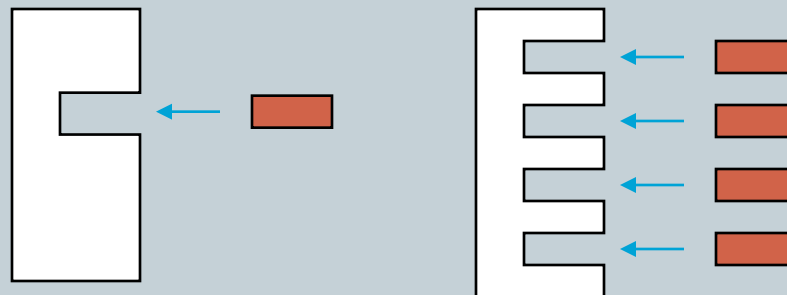
Exemples d'utilisation

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<!NOTATION GIF89a PUBLIC "-//Comuserve//NOTATION  
Graphics Interchange format 89a//EN" >
```

Les entités

31

- Sont des alias associant un nom à des « unités d'information »
- Les entités spécifiques au document sont décrites dans sa DTD interne
- Les entités plus générales sont décrites dans des DTD externes
- Chaque entité
 - ✦ est identifiée par un nom
 - ✦ est définie par une déclaration d'entité
 - ✦ est utilisée en appelant une référence d'entité



Utilisation des entités

32

- **Quand l'information**
 - Est utilisée dans plusieurs endroits
 - ✦ Ex. déclaration légale, caractère spécial
 - Est une partie d'un document qui doit être tronçonné pour rester gérable
 - ✦ Ex. livre = 1 fichier général + n chapitres soit n+1 fichiers en tout
 - Est conforme à un format de données différent de XML
 - ✦ Ex. : image JPEG

Types d'entités

33

- Entités internes
 - générales
 - ✦ utilisées dans les documents XML
 - « paramètre »
 - ✦ utilisées dans les déclarations dans les DTD
- Entités externes
 - générales
 - paramètres
- Entités analysables
- Entités non analysables
- Entités caractères
 - déjà vues

Entités générales internes

34

```
<!ENTITY nom "chaîne de remplacement">
```

- Entités analysables utilisées uniquement dans le document
- Référence : `&nom_entité;`
- Exemple

- Déclaration dans la DTD

```
<!ENTITY ENSSIB "École Nationale Supérieure des  
Sciences de l'Information et des Bibliothèques">
```

- Utilisation

```
<p>L'ENSSIB (&ENSSIB;) vient de publier...</p>
```

Entités générales externes

35

```
<!ENTITY nom SYSTEM "URI" >
```

- Permet de construire un document XML à partir de plusieurs autres documents
- Référence : `&nom_entité;`
- Exemple

- Déclaration dans la DTD

```
<!ENTITY doc SYSTEM "http://toto.org/doc.xml" >
```

- Utilisation

```
<aide>Voir cette URL : &doc; </aide>
```

Entités paramètres internes

36

```
<!ENTITY % nom "caractères de remplacement" >
```

- Entités analysables uniquement utilisées dans les DTD
- Référence dans la DTD : (%nom_entité;) (parenthèses conseillées)
- Exemples

- Déclarations DTD

```
<!ENTITY % tout "ANY" >
```

```
<!ENTITY % common "(para|list|table)">
```

- Utilisations dans la DTD

```
<!ELEMENT paragraphe %tout; >
```

```
<!ELEMENT chapter ((%common;)*, section*)>
```

```
<!ELEMENT section (%common;)*>
```

Entités paramètres externes

37

```
<!ENTITY % nom SYSTEM "URI" >
```

- Pour construire une DTD complexe à partir d'autres DTD complémentaires
- Référence dans la DTD : %nom_entité;
- Exemple

- Déclaration dans la DTD

```
<!ENTITY % règles SYSTEM "http://toto.org/regles.dtd" >
```

- Utilisation dans la DTD

```
%règles;
```

Entités analysables

38

- Le texte de remplacement fait partie intégrale du document
 - les données sont analysées correctement par le parser XML
- Déclaration dans la DTD comme ENTITY
- Utilisation avec `&nom;` ou `%nom;`

Entités non analysables

39

```
<!ENTITY % nom SYSTEM "URI" NDATA notation >
```

- **Pour déclarer un contenu non XML dans un document XML**
 - fichier image, audio, *etc.*
- **Référence : &nom_entité; uniquement comme attribut de type ENTITY**
- **Exemple**

- **Déclaration DTD**

```
<!NOTATION TIFF SYSTEM "format TIFF">
```

```
<!ENTITY photo SYSTEM "photo.tif" NDATA TIFF>
```

```
<!ELEMENT pic EMPTY>
```

```
<!ATTLIST pic name ENTITY #REQUIRED>
```

- **Utilisation dans le document XML**

```
<pic name="photo" />
```

Déclarations de notations

40

```
<!NOTATION nom SYSTEM "URI" >
```

- **Pour**

- Identifier par un nom le format des entités non XML externes
- Définir les formats des données et les applications qui permettent de les traiter

- **Exemple**

```
<!NOTATION GIF SYSTEM "GIF" >
```

```
<!NOTATION GIF89a PUBLIC "-//CompuServe//NOTATION  
Graphics Interchange format 89a//EN" >
```

Construire une DTD

41

- **Non trivial : il faut éviter de se tromper**
 - Changer une DTD XML a des conséquences sur les documents qui la suivent
- **Ressemble à la création d'un schéma de base de données**
- **Il faut considérer**
 - Le problème de la granularité
 - La questions des attributs et des éléments
 - Les limitations inhérentes aux DTD

Limites des DTD

42

- XML est seulement une syntaxe
- XML ne porte pas de sémantique
- Uniquement description de structure
- Pas de types

- Un des moyens de pallier certains problèmes
 - XML-schema

Problèmes des DTD

43

- Une syntaxe de description non-XML, héritée de SGML
 - Oblige à apprendre un langage supplémentaire
 - Ne permet pas de manipuler les DTD avec des outils XML
- Pas assez de contraintes sur les données manipulées
 - Toute données est une chaîne de caractères
 - Impossible de
 - ✦ spécifier des types simples
 - ex. entiers, dates, etc.
 - ✦ spécifier des cardinalités simples
 - ex. « un ARTICLE aura entre 1 et 4 MOTS-CLE »
 - ✦ spécifier des contraintes simples
 - ex. entier positif

Exemple de DTD réelle : bibteX

```
<!ENTITY % opt_fields "year?, volume?, pages?, month?, url?,
abstract?, note?">
<!ENTITY % req_fields "author, title">
<!ENTITY % key_atts "key ID #REQUIRED">

<!ELEMENT bibtex-file
(article|inproceedings|book|techreport|phdthesis|unpublished|misc
)*>

<!ELEMENT key      (#PCDATA)>
<!ELEMENT author  (name+)>
<!ELEMENT name    (#PCDATA)>
<!ELEMENT title   (#PCDATA)>
<!ELEMENT booktitle (short, long?)>
<!ELEMENT short   (#PCDATA)>
<!ELEMENT long    (#PCDATA)>(suite)
<!ELEMENT year    (#PCDATA)>
<!ELEMENT volume  (#PCDATA)>
<!ELEMENT pages   EMPTY>
<!ATTLIST pages
  first NMTOKEN #REQUIRED
  last  NMTOKEN #REQUIRED>
<!ELEMENT month   (#PCDATA)>
<!ATTLIST month
  mtype (short | long) "short">
<!ELEMENT url     EMPTY>
<!ATTLIST url
  ftype (ps|pdf|html) #REQUIRED
  href  CDATA        #REQUIRED>
<!ELEMENT abstract (#PCDATA)>
```

```
(suite)
<!ELEMENT number   (#PCDATA)>
<!ELEMENT note     (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT institution (#PCDATA)>
<!ELEMENT school   (#PCDATA)>
<!ELEMENT howpublished (#PCDATA)>
<!ELEMENT address  (#PCDATA)>
<!ELEMENT inproceedings (%req_fields;, booktitle,
%opt_fields;)>
<!ATTLIST inproceedings %key_atts;>
<!ELEMENT article  (%req_fields;, journal,
%opt_fields;)>
<!ATTLIST article   %key_atts;>
<!ELEMENT book     (%req_fields;, publisher,
%opt_fields;)>
<!ATTLIST book      %key_atts;>
<!ELEMENT techreport (%req_fields;, institution?,
number?, %opt_fields;)>
<!ATTLIST techreport %key_atts;>
<!ELEMENT phdthesis (%req_fields;, school?,
%opt_fields;)>
<!ATTLIST phdthesis %key_atts;>
<!ELEMENT unpublished (%req_fields;, %opt_fields;)>
<!ATTLIST unpublished %key_atts;>
<!ELEMENT misc      (%req_fields;, howpublished,
%opt_fields;)>
<!ATTLIST misc      %key_atts;>
```

(suite)

```
<!ELEMENT abstract    (#PCDATA)>
<!ELEMENT number     (#PCDATA)>
<!ELEMENT note       (#PCDATA)>
<!ELEMENT publisher   (#PCDATA)>
<!ELEMENT institution (#PCDATA)>
<!ELEMENT school     (#PCDATA)>
<!ELEMENT howpublished (#PCDATA)>
<!ELEMENT address    (#PCDATA)>
<!ELEMENT inproceedings (%req_fields;, booktitle, %opt_fields;)>
<!ATTLIST inproceedings %key_atts;>
<!ELEMENT article     (%req_fields;, journal, %opt_fields;)>
<!ATTLIST article     %key_atts;>
<!ELEMENT book        (%req_fields;, publisher, %opt_fields;)>
<!ATTLIST book        %key_atts;>
<!ELEMENT techreport  (%req_fields;, institution?, number?, %opt_fields;)>
<!ATTLIST techreport  %key_atts;>
<!ELEMENT phdthesis   (%req_fields;, school?, %opt_fields;)>
<!ATTLIST phdthesis   %key_atts;>
<!ELEMENT unpublished (%req_fields;, %opt_fields;)>
<!ATTLIST unpublished %key_atts;>
<!ELEMENT misc        (%req_fields;, howpublished, %opt_fields;)>
<!ATTLIST misc        %key_atts;>
```

Exercice 1

46

- On veut représenter en XML la feuille de présence à ce cours
- Cette feuille contiendra des données concernant
 - Le cours : date, heure, salle, UE
 - Les personnes présentes : enseignant, étudiants
- Proposez une DTD
- Écrivez la feuille de présence associée

Exercice 2

47

- Proposez deux DTD permettant de valider le document XML suivant

Un examen a entre 3 et 4 questions et chaque question a une ou plusieurs parties. Une partie se découpe en textes mélangés avec d'autres parties. Le code est uniquement alphanumérique, sans espace, la valeur de mois doit être une chaîne de caractère valide.

```
<examen code="coursXML">
<titre>Outils et documents XML</titre>
<date mois="jan" annee="2006"/>
<questions>
<question> <partie>Première partie<partie>Une sous-partie</partie>
</partie></question>
<question> <partie>Deuxième partie</partie> </question>
<question> <partie/> </question>
</questions>
</examen>
```