

Structuration de l'information XML

TD1 : DTD

Exercice 1 : une DTD d'étudiant

Donner la DTD d'un étudiant avec les renseignements suivants :

- son nom : unique et obligatoire
- son prénom : au moins un
- son âge : unique et obligatoire
- son adresse : unique et obligatoire
- sa classe : unique et obligatoire

Exercice 2 : un exemple XML d'étudiant

Ecrire un fichier etudiant.xml contenant un étudiant tel que vous l'avez décrit à l'exercice 1

Exercice 3 : validation

Valider le format de votre fichier XML en le confrontant à sa DTD à l'aide de l'outil validXML :

- en incluant la DTD dans le fichier XML
- en écrivant la DTD dans un fichier etudiant.dtd :
 - o en liant le fichier XML à sa DTD
 - o avec une simple confrontation sans indication dans le fichier XML

Exercice 4 : une DTD d'étudiants

Donner la DTD d'une liste d'étudiants (éventuellement vide). On reprend pour chaque étudiant la description de l'exercice 1 en ajoutant pour chacun la liste de ses matières (non vide). Pour chaque matière, on aura :

- o son libellé : unique et obligatoire
- o son évaluation : facultative avec
 - une note : obligatoire
 - un commentaire : facultatif

Exercice 5 : un exemple XML d'étudiants

Ecrire un fichier etudiants.xml contenant une liste telle que vous l'avez décrite à l'exercice 4

Exercice 6 : validation

Valider le format de votre fichier XML en le confrontant à sa DTD à l'aide de l'outil msv.

Exercice 7 : ABC

Ecrire une DTD **abc.dtd** qui décrit un document avec :

- un élément a en racine ;
- cet élément a a pour fils uniquement des éléments b et c en nombre et ordre quelconque ;
- les éléments b et c sont vides.

Exemple : `<a><c /><c /><c />`

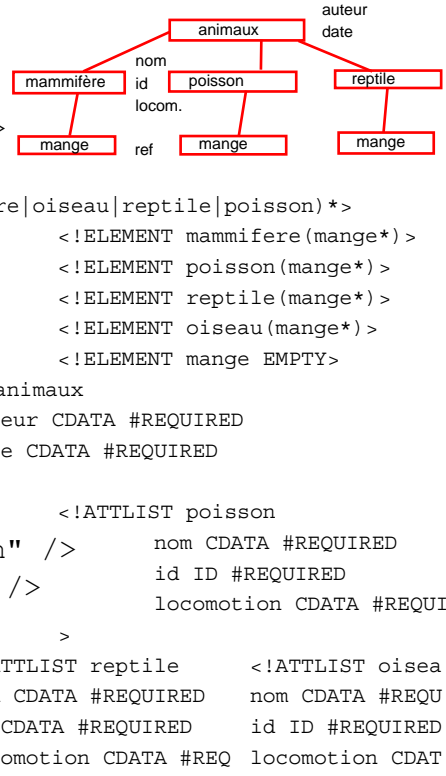
Exercice 8 : animaux

On décrit un ensemble d'animaux de manière simplifiée sous le format suivant :

```

<animaux auteur="Matthias Colin" date="2011-06-27">
  <mammifere nom="girafe" id="a1" locomotion="marche" />
  <mammifere nom="ours" id="a2" locomotion="marche" >
    <mange ref="a3" />
  </mammifere>
  <poisson nom="saumon" id="a3" locomotion="nage" />
  <mammifere nom="dauphin" id="a4" locomotion="nage" />
  <oiseau nom="aigle royal" id="a5" locomotion="vol" >
    <mange ref="a7" />
    <mange ref="a8" />
  </oiseau>
  <reptile nom="crocodile" id="a6" locomotion="nage" />
  <reptile nom="couleuvre" id="a7" locomotion="reptation" />
  <mammifere nom="marmotte" id="a8" locomotion="marche" />
  <mammifere nom="lion" locomotion="marche" >
    <mange ref="a1" />
  </mammifere>
</animaux>

```



Cet exemple cite toutes les catégories d'animaux et moyens de locomotion prévus. Ecrire une DTD **animaux.dtd** qui décrit un document tel que celui proposé.

Exercice 9 : ABE

- *: 0 ou n éléments +: Au moins un élément
- |: suivi optionnellement
- ,: suivi obligatoirement

On considère 3 possibilités de définition de l'élément e :

1. <!ELEMENT e(a*|b*)*> <- Autant de a qu'on veut suivis d'autant de b qu'on veut, répété n fois.
2. <!ELEMENT e(a*b*)> <- Autant de a qu'on veut suivis d'autant de b qu'on veut, une fois.
3. <!ELEMENT e(a,b)*> <- a ET b, répété n fois.

Les éléments a et b sont définis comme suit :

```

<!ELEMENT a EMPTY>
<!ELEMENT b EMPTY>

```

Pour chaque donnée XML, cocher la (ou les) DTD(s) qui le valide(nt) :

- | | | |
|--|--|-----------|
| 1. <e><a /><a /><a /></e> | <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 | <-1 et 3 |
| 2. <e><a /><a /><a /><a /><a /></e> | <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 | <-1 |
| 3. <e><a /><a /><a /></e> | <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 | <-1, 2 |
| 4. <e /> | <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 | <- 1 et 2 |

Exercice 10 : ABC

Donner une DTD qui accepte les données a) et b) et rejette la donnée c)

- | | |
|-------------------------------------|----------------------------|
| a) <a><c /> | <!ELEMENT e (b*, c+, b*) > |
| b) <a><c /><c /> | <!ELEMENT b EMPTY> |
| c) <a> | <!ELEMENT c EMPTY> |

Exercice 11 : ABCDE

Donner une DTD qui accepte les données a) et b) et rejette les données c) et d) :

- | | |
|---|-------------------------------------|
| a) <a><c /><e /> | <!ELEMENT a (b, d*, e*, b*, c, e) > |
| b) <a><d /><d /><e /><c /><e /> | <!ELEMENT a ((b d e)*, b, c, e) > |
| c) <a><e /> | |
| d) <a><c /><c /><e /> | |