

# Modélisation et interopérabilité :

## Semaine 39, TD-1

Benoît Valiron <[benoit.valiron@monoidal.net](mailto:benoit.valiron@monoidal.net)>

<http://inf356.monoidal.net/>

# Le Métalangage XML

# Exercice : Bien formé ? (6 choses)

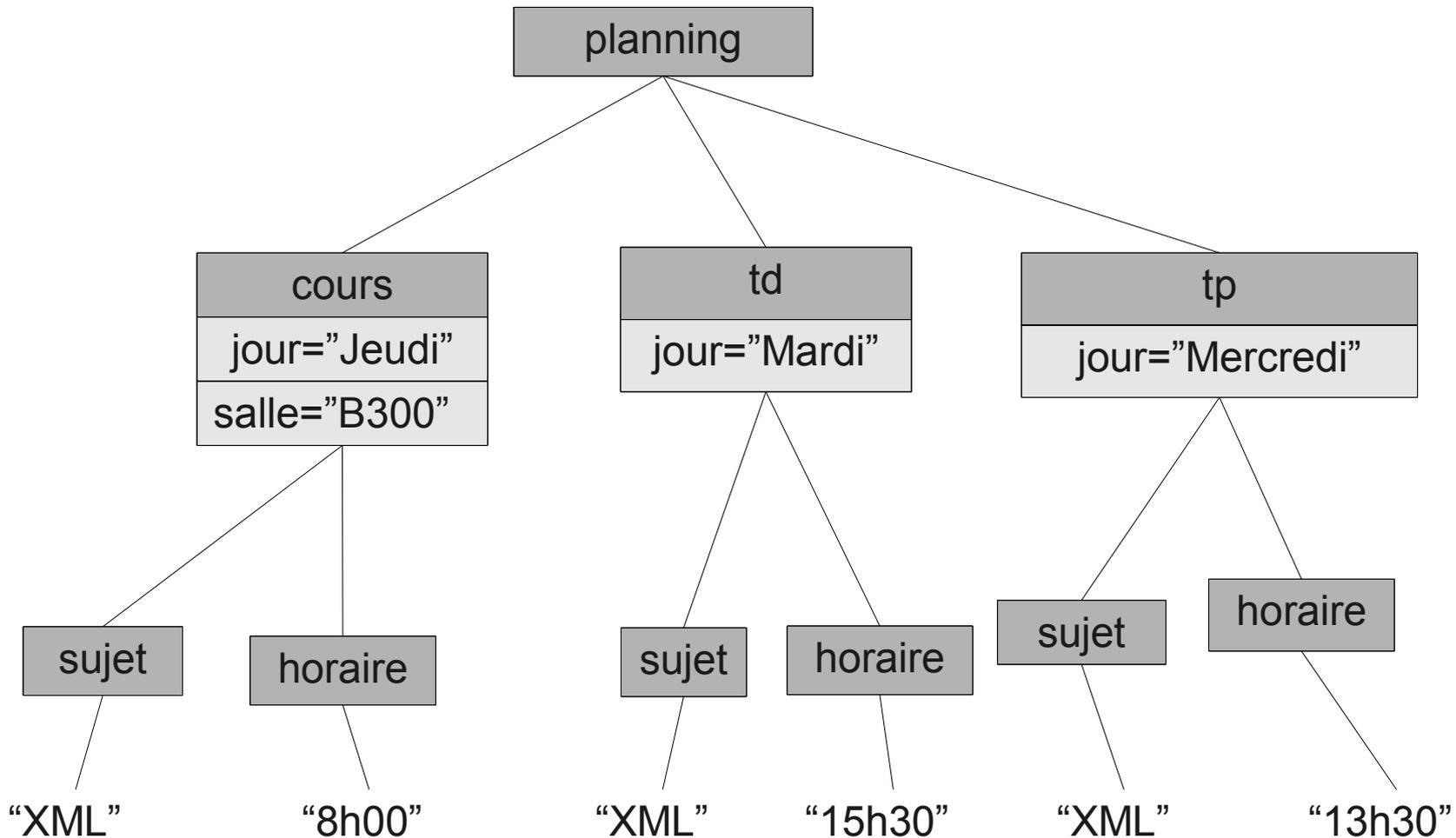
```
<?xml encoding="utf-8" version="1.0"?>
<cours jour = "Jeudi" salle="B300">
  < sujet>XML</sujet>
  <Horaire>8h00</horaire>
</cours>
<td jour=Mardi>
  <sujet>XML</sujet>
  <horaire>15h30</horaire>
<tp jour="Mercredi">
  <sujet>XML</sujet>
  <horaire>13h30</horaire>
</TP>
```

# Exercice : Bien formé ?

```
<?xml version="1.0" encoding="utf-8"?>
<planning>
  <cours jour = "Jeudi" salle="B300">
    < sujet>XML</sujet>
    <Horaire>8h00</horaire>
  </cours>
  <td jour="Mardi">
    <sujet>XML</sujet>
    <horaire>15h30</horaire>
  </td>
  <tp jour="Mercredi">
    <sujet>XML</sujet>
    <horaire>13h30</horaire>
  </TP>
</planning>
```

Arbre ?

# Arbre correspondant



# Lesquels sont des noms de balise corrects ?

1. `<123>`
2. `<_a>`
3. `<a:2>`
4. `<a.b.c>`
5. `<a-b-c>`
6. `<ASDF>`
7. `<a2:eee>`
8. `<cc/dd/ee>`
9. `<__>`
10. `<_:-->`
11. `<(aaa)>`
12. `<_aaa-a:_->`
13. `<dd;ee>`
14. `<1qwerty>`
15. `<aa __>`

# Validation DTD

# Feuille de style DTD

- Règles portant sur les éléments et les attributs
- Format:  

```
<!mot-clé paramètre1 paramètre2 ... >
```
- Mots clés: ELEMENT ATTLIST ENTITY (NOTATION)
  - ELEMENT : éléments XML
  - ATTLIST: attribut XML
  - ENTITY : caractères spéciaux et macro texte

**ELEMENT**

- Format :
  - <!ELEMENT nom ANY>
  - <!ELEMENT nom EMPTY>
  - <!ELEMENT nom (#PCDATA)>
  - <!ELEMENT nom (modèle\_de\_contenu)>
- Modèle de contenu (sans texte) est construit à l'aide de la syntaxe:
  - (a, b, ...) : liste ordonnée
  - (a | b | ...) : un élément au choix
  - (a\*) : zero, un ou plus
  - (a+) : un ou plus
  - (a?) : zero ou un
- Contenu mixte: (#PCDATA | a | b | ...)\*

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE personne [
<!ELEMENT personne (prénom, nom)>
<!ELEMENT prénom (#PCDATA) >
<!ELEMENT nom (#PCDATA) >
]>
<personne>
  <nom>Turing</nom>
  <prénom>Alan</prénom>
</personne>
```

**NON VALIDE...**

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE personne [
<!ELEMENT personne (prénom | nom)*>
<!ELEMENT prénom (#PCDATA) >
<!ELEMENT nom (#PCDATA) >
]>
<personne>
  <nom>Turing</nom>
  <prénom>Alan</prénom>
</personne>
```

**VALIDE !**

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE personne [
<!ELEMENT personne (prénom, nom)>
<!ELEMENT prénom (#PCDATA) >
<!ELEMENT nom (#PCDATA) >
]>
<personne>
  <prénom>Alan</prénom>
  <nom>Turing</nom>
</personne>
```

**VALIDE !**

# Exercice : <!ELEMENT tic ???>

- a- <tic><tac /><toc /></tic>
  - b- <tic><toc /><tac /></tic>
  - c- <tic><tac><toc /></tac><tac/></tic>
  - d- <tic><toc /></tic>
  - e- <tic><Toc /><toc /></tic>
  - f- <tic><toc /><toc /><tac /><tac></tac></tic>
- 
- 1- (tac, toc)
  - 2- (tac, toc)\*
  - 3- (tac\*, toc\*)
  - 4- (tac | toc)
  - 5- (tac | toc)\*
  - 6- (tac\* | toc\*)
  - 7- toc?
  - 8- tac?
  - 9- tac\*
  - 10- (tac? | toc\*)
  - 11- (tac | toc)?
  - 12- (tac+ | toc\*)

# ATTLIST

# ATTLIST

- Format :

```
<!ATTLIST elt attribut type #REQUIRED>
```

```
<!ATTLIST elt attribut type #IMPLIED>
```

```
<!ATTLIST elt attribut type #FIXED valeur>
```

```
<!ATTLIST elt attribut type valeur>
```

- REQUIRED : Attribut obligatoire
- IMPLIED : Attribut optionnel
- FIXED valeur : optionnel, mais fixé si présent
- valeur : optionnel, avec valeur par défaut

# Les types d'attributs

- Dix types d'attributs existent dans XML:
  - CDATA : texte quelconque
  - ENUMERATION : une liste d'unités lexicales
  - ID : **nom** XML unique dans le document
  - IDREF : référence à un attribut de type ID
  - IDREFS : “id1 id2 id3 ...”
  - NMTOKEN, NMTOKENS, ENTITY, ENTITIES, NOTATION.

# Énumérations

- Une énumération est sous la forme

```
<!ATTLIST date jour (Lundi | Mardi |  
Mercredi | Jeudi | Vendredi | Samedi |  
Dimanche) #REQUIRED>
```

- Les noms dans l'énumération sont des “unités lexicales nominales XML” : des lettres, des chiffres, \_ - . : sans restrictions.
- Exemple:

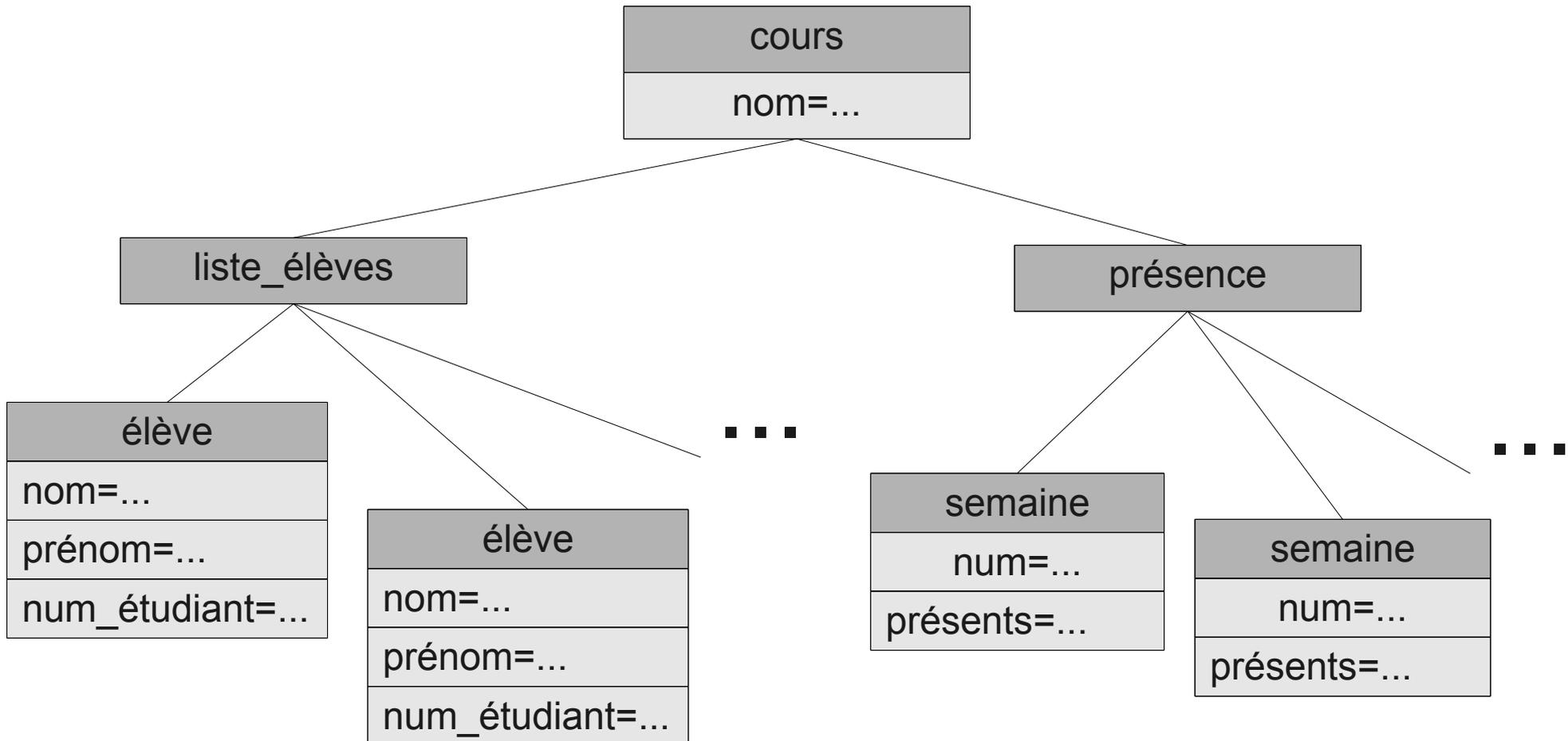
```
<!ATTLIST element att  
  (.bashrc | 123 | _.-ee | 3-4 | xml)  
  #IMPLIED>
```

# Exemple

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE organigramme [
<!ELEMENT organigramme (personne*,projet*)>
<!ELEMENT personne EMPTY>
<!ELEMENT projet EMPTY>
<!ATTLIST personne nom CDATA #REQUIRED>
<!ATTLIST personne id ID #REQUIRED>
<!ATTLIST personne vacataire (oui | non) "oui">
<!ATTLIST projet membres IDREFS #REQUIRED
      nom CDATA "Boire du café">
]>
<organigramme>
  <personne id="n1" nom="Bob"/>
  <personne id="n2" nom="Marie"/>
  <personne id="n3" nom="Alfred"/>
  <projet membres="n1 n2" nom="repeindre le couloir" />
  <projet membres="n1 n2 n3" />
  <projet membres="n3" nom="Truc urgent" />
</organigramme>
```

# Exercice

Donner une DTD correspondant à l'arbre suivant, et un exemple de document XML valide.



On utilisera une énumération pour les numéros de semaines (39 → 50) et des ID/IDREF pour les numéros d'étudiants et les présents.

# Solution

```
<!ELEMENT cours (liste_élèves, présence)>
<!ATTLIST cours nom CDATA #REQUIRED>
<!ELEMENT liste_élèves (élève+)>
<!ELEMENT élève EMPTY>
<!ATTLIST élève nom CDATA #REQUIRED
           prénom CDATA #IMPLIED
           num_étudiant ID #REQUIRED>
<!ELEMENT présence (semaine+)>
<!ELEMENT semaine EMPTY>
<!ATTLIST semaine num
           (38 | 39 | 40 | 41 | 42 |
            43 | 44 | 45 | 46 | 47 |
            48 | 49 | 50) #REQUIRED>
<!ATTLIST semaine présents IDREFS #REQUIRED>
```

# Un exemple de document

```
<cours nom="INF101">
  <liste_élèves>
    <élève nom="Turing"
      prénom="Alan"
      num_étudiant="n0123" />
    <élève nom="Church"
      prénom="Alonzo"
      num_étudiant="n0124" />
    <élève nom="Clark"
      prénom="David"
      num_étudiant="n0002" />
  </liste_élèves>
  <présence>
    <semaine num="39" présents="n0123 n0124" />
    <semaine num="40" présents="n0123 n0002" />
    <semaine num="41" présents="n0124 n0123 n0002" />
  </présence>
</cours>
```

ENTITY

# ENTITY

- Deux types d'entités majeurs :
  - Les entités générales
  - Les entités paramètres
- Entités générales : pour faire des abréviations  
`<!ENTITY moi "Benoît Valiron">`
- Utilisés comme `&apos;`, `&lt;`, `&lq;`, `&amp;`, `&quot;`;  
`<texte>Je m&apos;appelle &moi;</texte>`
- Cas particulier : `&#xxx;` signifie le caractère de valeur xxx dans le jeu de caractères

# Exemple

```
<?xml version="1.0"?>
<!DOCTYPE texte [
<!ELEMENT texte (#PCDATA)>
<!ENTITY b1 "Tic ">
<!ENTITY b2 "&b1; &b1;">
<!ENTITY b3 "&b2; &b2;">
<!ENTITY b4 "&b3; &b3;">
]>
<texte>&b4;</texte>
```

# Contre-exemple

```
<?xml version="1.0"?>
<!DOCTYPE plan [
<!ENTITY pt "(x, y, z)">
<!ELEMENT plan (&pt;, &pt;, &pt;)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
<!ELEMENT z (#PCDATA)>
]>
<plan>
<x>0</x><y>0</y><z>0</z>
<x>0</x><y>1</y><z>0</z>
<x>0</x><y>0</y><z>1</z>
</plan>
```

**NON-VALIDE !**

# ENTITY (suite)

- Entités paramètre : utilisable uniquement dans la DTD

```
<!ENTITY % nom_entité texte_replacement>
```

- Pour des raccourcis. Exemple:

```
<!ENTITY % liste "(a | b | c | d) #REQUIRED">  
<!ELEMENT alphabet EMPTY>  
<!ATTLIST alphabet lettre %liste;>
```

- Définition et utilisation en DTD externe
- Redéfinition possible en DTD interne

# Exemple

## tic.dtd

```
<?xml version="1.0" encoding="utf-8"?>
<!ENTITY % tic '"Tic "'>
<!ELEMENT texte (#PCDATA)>
<!ENTITY b1 %tic;>
<!ENTITY b2 "&b1; &b1;">
<!ENTITY b3 "&b2; &b2;">
<!ENTITY b4 "&b3; &b3;">
```

## fichier.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE texte SYSTEM "tic.dtd" [
<!ENTITY % tic '"Tac "'>
]>
<texte>&b4;</texte>
```

# Exercice

triangle.dtd

```
<!ENTITY % pt "(x, y, z)">
<!ELEMENT triangle (%pt;, %pt;, %pt;)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
<!ELEMENT z (#PCDATA)>
```

volume.xml

```
<?xml version="1.0"?>
<!DOCTYPE triangle SYSTEM "triangle.dtd">
<triangle>
<x>0</x><y>0</y><z>0</z>
<x>0</x><y>1</y><z>0</z>
<x>0</x><y>0</y><z>1</z>
</triangle>
```

Question : Écrire un fichier plan.xml avec triangle.dtd.

# Solution

plan.xml

```
<?xml version="1.0"?>
<!DOCTYPE triangle SYSTEM "triangle.dtd" [
<!ENTITY % pt "(x, y)">
]>
<triangle>
<x>0</x><y>0</y>
<x>0</x><y>1</y>
<x>0</x><y>0</y>
</triangle>
```

# Exercice : <!ELEMENT tic ???>

1. (a<sup>\*</sup>)

2. (a<sup>+</sup> | a<sup>\*</sup>)

3. (a, a<sup>\*</sup>)

4. (a, a<sup>+</sup>)

5. (a<sup>\*</sup>, (a | (a, a)))

6. (a, a)<sup>\*</sup>

7. (a, (a, a))<sup>\*</sup>

8. (a | (a, a<sup>+</sup>))

9. (a | a)

10. (a?, b<sup>+</sup>)

11. ((a<sup>+</sup>, b<sup>+</sup>) | (a?, b?))

12. (a, b)<sup>+</sup>

13. (a | b)<sup>+</sup>

14. ((a, b) | (b, a))

15. ((a | b), (a | b))

16. ((a, b) | (b?, a?))

17. ((a? | b?), (a? | b?)?)