

XML et Données Semi Structurées

Document Type Definition
DTD

Document Type Definition

Qu'est-ce qu'une DTD ?

- Une DTD décrit de manière précise les éléments que peut contenir un document XML, dans quel ordre ils peuvent apparaître et quels sont leurs attributs possibles.
- Elle permet donc de vérifier qu'un document XML est conforme à une syntaxe donnée (à une grammaire).
- La vérification de la conformité d'un document XML se fait dans une DTD (Document Type Definition)

Document Type Definition

On distingue 2 types de conformité:

☐ Les documents bien formés: les documents XML ne comportant pas de DTD mais répondant aux règles de base de XML

☐ Les documents valides : les documents XML avec une DTD

Document Type Definition

Une DTD peut être définie de 2 façons :

- sous forme interne, en incluant la grammaire dans le document
- sous forme externe, soit en appelant un fichier la grammaire à partir d'un fichier local ou bien en y accédant par URL.

Document Type Definition

- Types de déclaration dans une DTD: une DTD contient des déclarations (conformes à la syntaxe indiquée) pour:

- les éléments:

```
<!ELEMENT element-name content-model >
```

- Les attributs:

```
<!ATTLIST element-name attrib-name type default>
```

- Les entités:

```
<!ENTITY entity-name "replacement text">
```

Document Type Definition

Les notations:

```
<!NOTATION notation_name SYSTEM "text">
```

- Les notations sont employées dans les types d'attribut de la DTD pour identifier le mode d'encodage des données analysées ou non analysées.
- Elles sont habituellement utilisées par les applications capables de traiter les données.
- **NB:** Les notations sont rarement employées dans les DTD.

Document Type Definition

- **Exemple de déclaration d'éléments:**

```
<!ELEMENT employees (employee)>  
<!ELEMENT employee (name)>  
<!ELEMENT name (#PCDATA)>
```

- **Exemple d'un document XML basé sur la DTD:**

```
<?xml version="1.0"?>  
<employees>  
  <employee>  
    <name>Steven King</name>  
  </employee>  
</employees>
```

Document Type Definition

Comment référencer une DTD ?

Dans un document XML, la DTD est référencée:

- après la déclaration XML et avant la racine en utilisant:

```
<!DOCTYPE employees [ ... ]>
```

- de façon externe avec les mots-clés SYSTEM ou PUBLIC

```
<!DOCTYPE employees SYSTEM "employees.dtd">
```

```
<!DOCTYPE employees PUBLIC "-//formal-public-ID">
```


Document Type Definition

- **Exemple de document XML avec une DTD interne**

```
<?xml version="1.0"?>
<!DOCTYPE employees [
<!ELEMENT employees (#PCDATA)>
]>
<employees>Employee Data</employees>
```

- **Remarque :** utilisez le nom de l'élément racine après <!DOCTYPE

Document Type Definition

La spécification `<!DOCTYPE name [...]>` apparaît après la déclaration XML et avant l'élément racine du document XML.

Un document XML peut référencer à la fois une DTD externe et une DTD interne. Si un élément est défini à la fois en externe et en interne, la DTD interne a la priorité.

Document Type Definition

Déclarations d'éléments

- Syntaxe de déclaration d'un élément :

```
<!ELEMENT element-name content-model>
```

- Quatre type de modèle de contenu

1

```
<!ELEMENT job EMPTY>
```

2

```
<!ELEMENT employees (employee)>  
<!ELEMENT employee (employee_id,last_name,job_id)>  
<!ELEMENT job_id (manager | worker)>
```

Document Type Definition

Déclarations d'éléments

3

```
<!ELEMENT employee_id ANY>
```

4

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

```
<!ELEMENT hire_date (date | (day,month,year))>
```

```
<!ELEMENT person (#PCDATA | first_name | phone)*>
```

Explication de la syntaxe: Une déclaration d'élément contient le mot-clé **ELEMENT**, suivi du **nom de l'élément** et d'un **modèle de contenu**.

Le modèle de contenu définit ce que peut contenir l'élément.

Document Type Definition

Cardinalité des éléments

Le symbole de cardinalité

- Indiquent le nombre d'éléments enfant autorisés
- Apparaissent sous la forme de suffixes:

Aucun symbole (par défaut)	Obligatoire (un et un seul)
? (point interrogation)	Zéro ou un (facultatif)
* (astérisque)	Zéro ou plus (facultatif)
+ (signe plus)	Un ou plus (obligatoire)

Document Type Definition

Cardinalité des éléments

Le modèle de contenu indique comment regrouper les sous-éléments à l'aide de parenthèses, ainsi que le nombre d'occurrence autorisés pour chaque éléments, ou cardinalité.

La cardinalité s'exprime à l'aide des symboles suivants:

- point d'interrogation (?)
- astérisque (*)
- signe plus (+).

Document Type Definition

Cardinalité des éléments

Dans la spécification du modèle de contenu, les symboles peuvent apparaître en tant que suffixes avec

- chaque élément,
- un groupe d'éléments
- le modèle de contenu complet.

Par défaut, **l'absence de symbole** signifie que l'élément, le groupe ou le modèle de contenu doit apparaître **au moins une fois, obligatoirement**.

Document Type Definition

Cardinalité des éléments

Exemple d'utilisation de symbole:

1. Un élément <employees> peut contenir zéro ou plusieurs <employee>:

<!ELEMENT employees (employee*)>

2. Un élément <department> doit au moins contenir un élément <employee> :

<!ELEMENT department (employee+)>.

Document Type Definition

Cardinalité des éléments

3. Un élément `<employee>` doit avoir un élément `<last_name>` et un élément `<salary>` , ainsi qu'un élément facultatif `<commission_pct>` :

`<!ELEMENT employee(last_name, salary, commission_pct ?)>`

4. L'élément `<jobs>` doit avoir un président et facultativement, plusieurs managers ou employés:

`<!ELEMENT jobs (president, (manager | clerk)*)>`

Document Type Definition

Déclaration d'attributs

Les attributs fournissent des informations supplémentaires sur un élément et constituent une paire (nom-valeur) décrivant une propriété de l'élément.

La déclaration d'un attribut nécessite:

- un nom d'élément qui identifie l'élément auquel appartient l'attribut
- un nom d'attribut
- une valeur littérale placée entre guillemets qui est utilisée comme valeur par défaut lorsque le document XML n'en mentionne pas d'autre

Document Type Definition

Déclarations d'attributs

- Syntaxe de déclaration d'un attribut

```
<!ATTLIST element-name attrib-name type default>
```

- Nécessite:
 - **un nom d'élément** (*element-name*)
 - **un nom d'attribut** (*attrib-name*)
 - **un type d'attribut** indiqué par : CDATA, énumération, ENTITY, ENTITIES, ID, IDREF, IDREFS, NMTOKEN, NMTOKENS et NOTATION

Document Type Definition

Déclarations d'attributs

- un type d'attribut par défaut indiqué par : #IMPLIED, #REQUIRED, #FIXED ou une valeur littérale

Exemple

```
<!ELEMENT employee (employee_id, last_name)>
```

```
<!ATTLIST employee manager_id CDATA #IMPLIED>
```

Document Type Definition

Déclarations d'attributs

Exemple de document XML avec la déclaration d'attribut **manager_id**:

```
<employee manager_id="100">  
  <employee_id>101</employee_id>  
  <last_name>Kochhar</last_name>  
</employee>
```

Document Type Definition

Types d'attribut CDATA et Enumération

- **CDATA** : pour des valeurs de données **alphanumériques**

```
<!ELEMENT employee (employee_id, last_name)>  
<!ATTLIST employee manager_id CDATA #IMPLIED>
```

```
<employee manager_id="100">  
    <employee_id>101</employee_id>  
    <last_name>Kochhar</last_name>  
</employee>
```

- **NB**: La plupart des attributs sont de type **CDATA**. Le type de données CDATA indique que l'attribut peut prendre n'importe quelle valeur de caractère

Document Type Definition

Types d'attribut CDATA et Enumération

- **Enumération:** pour permettre un choix dans une liste de valeurs

```
<!ELEMENT employee (employee_id, last_name)>  
<!ATTLIST employee gender (male|female) #IMPLIED>
```

```
<employee gender="male">  
  <employee_id>104</employee_id>  
  <last_name>Ernst</last_name>  
</employee>
```

- **NB:** Le type d'attribut énuméré offre un choix dans une liste de valeurs autorisées pour l'attribut en question

Document Type Definition

Spécifier la valeur d'attribut par défaut

Elle est requise dans la DTD lorsqu'est utilisé le mot-clé

#FIXED

```
<!ELEMENT employee (employee_id, last_name)>  
<!ATTLIST employee manager_id CDATA #IMPLIED>  
<!ATTLIST employee min_salary CDATA #FIXED '4000'>
```

NB: une valeur d'attribut est obligatoire dans un document XML lorsqu'est utilisé le mot-clé **#REQUIRED**

Document Type Definition

Type d'attribut ID

ID: pour identifier de façon unique une instance d'élément

```
<!ELEMENT employees (employee*)>  
<!ELEMENT employee (last_name)>  
<!ATTLIST employee employee_id ID #REQUIRED>
```

```
<employees>  
  <employee employee_id="_103">  
    <last_name>Hunold</last_name>  
  </employee>  
</employees>
```

Document Type Definition

Type d'attribut ID

Le type **ID** permet d'identifier de façon unique des éléments spécifiques d'un document XML.

Il est utile pour repérer un élément par sa valeur ID. pour qu'il soit possible d'utiliser un type ID:

- l'élément ne doit avoir qu'un seul attribut ID
- tous les éléments doivent avoir une valeur différente (unique) pour l'attribut ID

Document Type Definition

Type d'attribut ID

- Sa valeur doit correspondre à un nom XML valide. Celui-ci doit commencer par un caractère alphanumérique ou un caractère de soulignement, jamais un caractère numérique.

NB: l'utilisation de la valeur d'attribut #REQUIRED par défaut n'est pas obligatoire, mais elle est recommandée pour les attributs déclaré avec le type ID

Document Type Definition

Types d'attribut **IDREF** et **IDREFS**

Les types **IDREF** et **IDREFS** permettent à d'autres éléments de référence un ou plusieurs éléments avec un attribut **ID unique**.

Déclarez le type d'attribut en tant que:

- type **IDREF** pour référencer **une valeur** d'attribut unique d'un élément
- type **IDREFS** pour référencer **plusieurs valeurs** d'attribut

Document Type Definition

Types d'attribut **NMTOKEN** et **NMTOKENS**

Le type **NMTOKEN** représente un jeton de nom, dont la valeur est semblable à un nom XML, c'est-à-dire qu'elle contient des valeurs alphanumériques, notamment:

des lettres (A-Z, a-z), des chiffres (0-9), des caractères de soulignements (_), des deux points (:), des tirets (-), des points (.)

NB: toutefois une valeur **NMTOKEN** ne peut pas contenir des caractères non imprimables

Document Type Definition

Types d'attribut NMTOKEN et NMTOKENS

```
<!ELEMENT employees (employee*)>
<!ELEMENT employee (last_name)>
<!ELEMENT last_name (#PCDATA)>
<!ATTLIST employee hire_date NMTOKEN #IMPLIED>
<!ATTLIST employee email NMTOKENS #IMPLIED>
```

```
<employees>
  <employee hire_date="13-JAN-1993">
    <last_name>De Haan</last_name>
  </employee>
  <employee email="AHUNOLD Alexander.Hunold">
    <last_name>Hunold</last_name>
  </employee>
</employees>
```

Document Type Definition

Types d'attribut NMTOKEN et NMTOKENS

Avec le type NMTOKENS , il est possible d'utiliser une liste de valeur d'attribut NMTOKEN séparées par un espace.

Document Type Definition

Entités en XML

Types d'entités prédéfinies par normes XML

- entités intégrées
- entités caractères

Les entités de la DTD peuvent être déclarées en tant que:

- entités générales
- entités paramètre

Document Type Definition

Entités en XML

Elles référencent:

- les entités intégrés, des entités caractère et des entités générales dans un document XML ou une DTD à l'aide

de: `&entity_name;`

- des entités paramètre dans la DTD à l'aide de:

`%entity_name;`

Document Type Definition

Déclaration d'entités

Entités analysables internes:

```
<!ENTITY entity-name "replacement text">
```

```
<!ENTITY company "Oracle Corporation">
```

```
<!ENTITY Emperor "Alexander &quot;The Great&quot;">
```

```
<!ENTITY president "<employee>  
                        <last_name>King</last_name>  
                        </employee>" >
```

Entités analysable externes:

```
<!ENTITY entity-name SYSTEM "file.ext | URL">
```

Document Type Definition

Déclaration d'entités paramétriques

- **Déclaration d'entités internes:**

```
<!ENTITY % entity-name "replacement text">
```

- **Déclaration d'entités externes**

```
<!ENTITY % entity-name SYSTEM "file.ext | URL">
```

NB: les déclarations doivent être faites dans une DTD externe.

Document Type Definition

Déclaration d'entités paramétriques

- **Exemple d'utilisation d'entité dans une DTD:**

```
<!ENTITY % employee_elements "last_name, salary">  
<!ENTITY % employee_elements SYSTEM "empelm.txt">
```

```
%entity-name;
```

```
<!ELEMENT employee (%employee_elements;)>
```

Exemple complet d'une DTD

```
<!ELEMENT employees (employee+)>
<!ELEMENT employee (employee_id, last_name)>
<!ATTLIST employee manager_id CDATA #IMPLIED
                  department_id (10|60|90) '90'>
<!ELEMENT employee_id (#PCDATA )>
<!ELEMENT last_name (#PCDATA )>
```

```
<?xml version="1.0"?>
<!DOCTYPE employees SYSTEM "employees.dtd" [
  <!ENTITY title "Mr">
]>
<employees>
  <employee manager_id="100" department_id="10">
    <employee_id>100</employee_id>
    <last_name>&title; king</last_name>
  </employee>
</employees>
```