Topics :

1. Définition
2. Déclaration
   1. Interne
   2. Externe
      1. Adressée par URL
      2. Adressée par FPI
3. Structure d’une DTD :
   1. Eléments
      1. Types d’éléments
         1. Elément vide
         2. Séquence d’éléments pur
         3. Elément textuel
         4. Elément mixte
         5. Elément libre
   2. Attributs
      1. Declaration
      2. types
   3. Entités

Pour qu’un Document XML soit valide il doit respecter un ensemble des règles qui définissent son structure, et pour décrire cette structure du document XML, on peut utiliser les DTD ou les XSD.

**1-Qu’est-ce qu’une DTD ?**

La DTD **(Document Type Definition )** est **un ensemble de règles** qui définissent les **éléments**, les **attributs** et les **entités** qui y sont autorisés ainsi que les relations entre éléments et attributs, leurs **ordres** et leurs **fréquences d’apparition** dans ce document.

**2-Déclaration :**

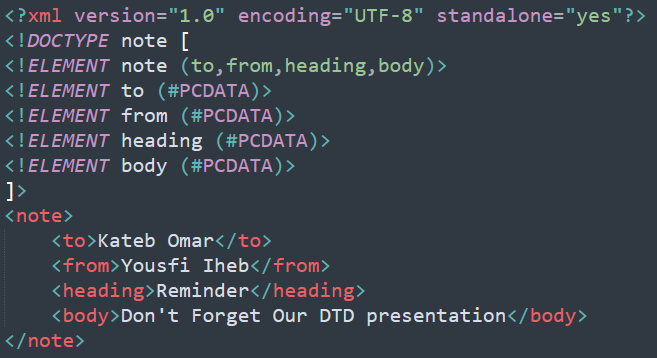
La DTD peut être déclaré dans un Document XML en deux formats

**2-1-interne :**

Lorsque la DTD est incluse directement dans le Document XML elle est appelé interne, elle suit la syntaxe suivante :

< !DOCTYPE **élément-racine** [Séquence des déclarations/instructions ]>

Exemple :



La DTD ci-dessus est interprétée comme suit :

• «<!DOCTYPE note [ : définit que l'élément racine de ce document est « **note** ».

• <!ELEMENT **note** (to, from, heading, body)> définit que l'élément « **note** » doit contenir quatre éléments : **«to, from, heading, body ».**

• <!ELEMENT **to** (#PCDATA)> définit que l'élément « **to** » doit être de type **« #PCDATA** **».**

• <!ELEMENT **from** (#PCDATA)> définit que l'élément « **from** » doit être de type « **#PCDATA** ».

• <!ELEMENT **heading** (#PCDATA)> définit que l'élément « **heading** » doit être de type « **#PCDATA** ».

• <!ELEMENT **body** (#PCDATA)> définit que l'élément « **body** » doit être de type « **#PCDATA** ».

**## PCDATA (Parsed Character Data) c’est-à-dire une chaine de caractères qui doit être parser par l’analyseur syntaxique .**

**2-2-externe :**

Lorsque la DTD est externe, celle-ci est contenue dans un autre fichier dont l'extension est généralement « **.dtd ».**

La déclaration se fait selon la syntaxe suivante :

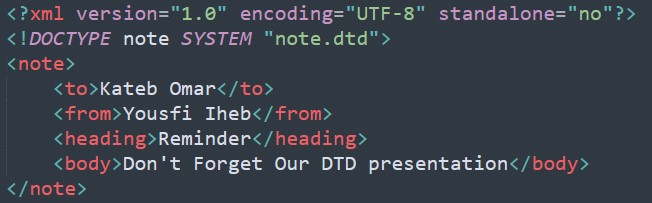
**2-2-1-Adressée par URL :**

<!DOCTYPE **element\_racine** SYSTEM "URL">

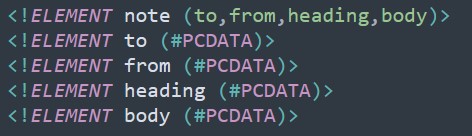
#URL peut être un lien complet ou le nom d’un fichier DTD local

Exemple:

Note.xml:



Note.dtd:



**2-2-2-Adressée par FPI :**

<!DOCTYPE **element\_racine** PUBLIC **FPI** "URI">

Les FPI (Formal Public Identifier) sont des identifiants de documents hérités de SGML. Ils sont plutôt remplacés en XML par les [URI](https://www.irif.fr/~carton/Enseignement/XML/Cours/Syntax/index.html#sect.syntax.uri) qui jouent le même rôle. Ils sont constitués de quatre parties séparées par des chaînes '//' et ils obéissent donc à la syntaxe suivante :

”[Registration]//[Organisation]//[Type][Label]//[Language] ”

Exemple :

< !DOCTYPE note PUBLIC ”-//W3C//DTD XHTML 1.0 Transitional//EN ” ”http://www.w3.org/TR/xhtml/DTD/xhtml-transitional.dtd ”>

**3-Structure d’une DTD :**

La DTD est construite à partir d’un ensemble de déclarations permettant de définir les noms, les contenus et les attributs des éléments qui composent le fichier XML

Chacune de ces déclarations commence par « < !  » Suivi d’un mot clé qui indique le type de déclaration :

* < !ELEMENT> déclaration des **éléments** : les éléments permettent de définir le contenu du fichier XML.
* < !ATTLIST> déclaration des **attributs** : les attributs permettent d’enrichir la sémantique des éléments.
* < !ENTITY> déclaration des **entités** : les entités permettent de définir des paramètres à l’intérieur de la DTD

La declaration se termine par «  >  »

**3-1-les éléments :** Les déclarations d'éléments constituent le cœur des DTD car elles définissent la structure des documents valides. Elles spécifient quels doivent être les enfants de chaque élément et l'ordre de ces enfants.

La déclaration d'un élément est nécessaire pour qu'il puisse apparaître dans un document XML

la déclaration d'un élément prend la forme suivante :

< !ELEMENT **élément** type >

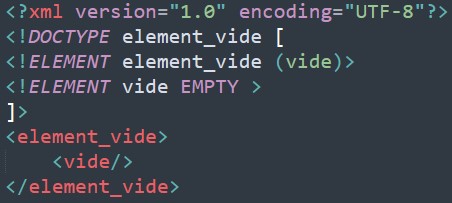
**3-1-1 types d’éléments :**

**#Elément vide :**

La déclaration suivante indique que le contenu de l'élément est nécessairement vide. Cet élément peut uniquement avoir des attributs. Les éléments vides sont souvent utilisés pour des liens entre éléments.

< !ELEMENT **élément** EMPTY>

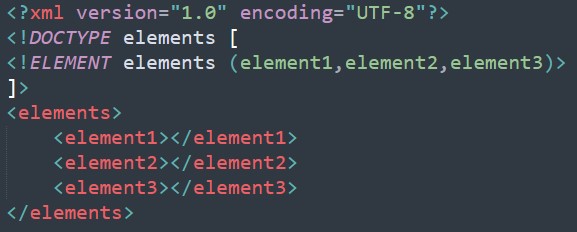
Exemple :



**#séquence d’éléments purs :**

La déclaration d'une séquence élément pur détermine quels sont les enfants d’éléments et dans quel ordre ils doivent apparaître.

Exemple :

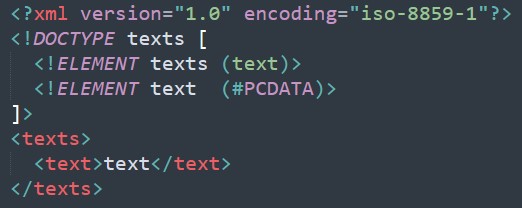


**#élément textuel :**

< !ELEMENT **élément** (#PCDATA)>

Cette déclaration indique qu'un élément peut uniquement contenir du texte.

Exemple :

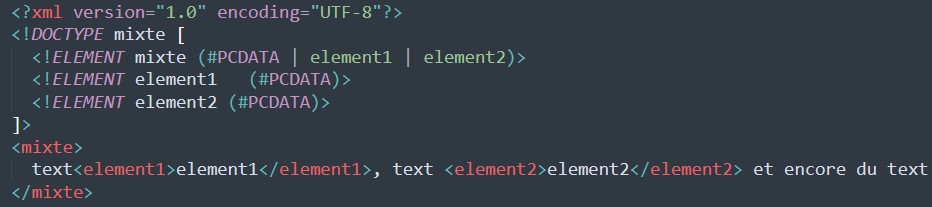


**#élément mixte :**

<!ELEMENT **élément** (#PCDATA | element1 | ... | élémentN)>

Cette déclaration indique qu’un élément peut contenir du texte ainsi que d’autre éléments

Exemple :



**#élément avec contenu libre :**

< !ELEMENT **élément** ANY>

Cette déclaration indique que l’élément peut contenir n’importe quel type de donnée.

***#### Remarque :***

Les opérateurs et les indicateurs d’occurrence :

|  |  |
| --- | --- |
| **Operateur** | **Signification** |
| **,** | Séquence |
| | | Choix |
| ? | 0/1 occurrence |
| \* | nombre quelconque occurrences |
| + | Minimum une occurrence |

3-2-les attributs :

3-2-1-Declaration des attributs :

Pour qu'un document soit valide, tout attribut présent dans la balise ouvrante d'un élément doit être déclaré. La déclaration d'attribut prend la forme générale suivante:

<!ATTLIST element **attribut** type default> où

élément : le nom de l'élément auquel il appartient.

attribut : est le nom de l'attribut.

Type : type d’attribut déclaré .

Default : la valeur par default d’attribut .