DTD (Document Type Definition): sert pour créer notre propre langage ou bien notre propre balise.

Syntaxe :

< !DOCTYPE racine[

< !Element element1 type1>

< !ELEMENT element2 type2>

< !ATTLIST element1 attribute1 attributType constraint

Attribute2 attributType2 constraint>

< !ATTLIST element2 attribute2 attributType constraint>

]>

Le type d’élément définit quoi l’élément va contenir :

* (#PCDATA) 🡪 élément contient que du texte.
* EMPTY 🡪 élément ne contient rien, comme une balise img, input…
* (ele1, ele2, (ele3|ele4),ele5+,ele6\*) 🡪 ou continent une séquence d’élément qui doit être respecter par ordre, ici ele3 ou ele4 qui va être comme 3-eme balise, élément 5 est obligatoire mais il peut être répéter plusieurs fois, et l’élément 6 est facultatif mais qui peut être répéter plusieurs fois.

Le type de l’attribut :

* CDATA : l’attribut contient que du texte.
* (text1|text2) : la valeur de l’attribut contient que text1 ou text2.
* ID : pour des valeur unique et NMTOKEN.
* IDREF : équivalent la clé étrangère.
* NMTOKEN : c’est un nom XML valide, qui commence par une lettre et qui ne contient ni espace ni caractère spéciale.

Le type de contraintes d’un attribut :

* #REQUIRED : l’attribut est obligatoire.
* #IMPLIED : l’attribut est facultatif.
* #FIXED ‘’VALEUR’’ : ici une valeur fixée qui est obligatoire.
* ‘’valeur1’’ pour exiger une valeur par défaut.

<!DOCTYPE liste [

<!ELEMENT nom (#PCDATA) > <!-- cet element va contenir du texte -->

<!ELEMENT prénom (#PCDATA) >

<!ELEMENT age (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ELEMENT photo EMPTY> <!-- element photo ne contient rien, du coup EMPTY -->

<!ELEMENT description (#PCDATA)>

<!ELEMENT CV EMPTY>

<!ELEMENT tél (#PCDATA)>

<!ELEMENT étudiant (nom,prénom,age,tél\*,email,photo,(description|CV)) >

<!-- element etudiant contient une sequece qui doit etre bien respectee-->

<!ELEMENT liste (étudiant)+> <!-- liste ici fait reference a la racine, qui indique ici

qu'elle va contenir un element etudiant ou plusieurs -->

<!ATTLIST tél type (fixe|portable) "portable"> <!-- attribut tel comme valeur portable ou fixe,

pardefaut portable-->

<!ATTLIST CV file CDATA #REQUIRED>

<!ATTLIST photo source CDATA #REQUIRED

                hauteur CDATA #IMPLIED

                largeur CDATA #IMPLIED

                >

<!ATTLIST étudiant code ID #REQUIRED>

<!ATTLIST liste année CDATA #FIXED "2021">

]>

**XSD (XML Sheema Definition)** : á le même fonctionnement que le DTD, mais il utilise le syntaxe XML, permet de définir des éléments locaux, et il support l’espace de nom.

En XSD un élément soient simple soit complexe, un élément simple c’est un élément qui contient que du texte et ne contient ni attribut ni autre élément.

Un document shema .xsd, est un document ou l’élément racine est schema et qui utilise obligatoirement le namespace <http://www.w3.org//2001/XMLSchema> .



Déclaration d’un élément :

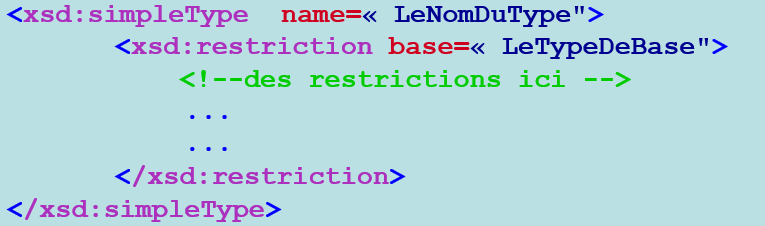
La déclaration se fait par l’élément xsd:element :



Le type peut être simple (prédéfinit) ou bien complexe(créer) :

Les types prédéfinit : string, integer, decimal, date, id, idref, boolean …

On peut créer un type simple basé sur un type prédéfinit avec le syntaxe suivant :



Attribut **name** utiliser pour spécifier le nom de nouveau type, et l’attribut **base** d’élément restriction utiliser pour spécifier le nom de type prédéfinit qu’on veut travailler sur.

Les type des restrictions possible :

<xsd :pattern value = ‘’ER’’ /> utiliser pour exiger une expression régulière.

<xsd :ennumeration value = ‘’P’’> utiliser plusieur fois pour exiger une liste des choix possible.

<xsd :minInclusive value = ‘’0’’> exiger que le nombre entrée soit supérieur ou égal à 0.

<xsd :minExclusive value = ‘’0’’> exiger que le nombre entrée soit supérieur stric à 0.

Même chose il y’a maxInclusive et maxExclusive.

<xsd :length value = ‘’10’’> la taille de la chaine doit être exactement 10.

<xsd :minLength value = ‘’4’’> la taille minimum de la chaine entre doit être supérieur a 10.

<xsd :maxLength value = ‘’32’’> la taille ne doit pas dépassée 32 caractères.

<xsd :totalDigits value =’’4’’> 4 chiffres qui seront afficher.

<xsd :fractionDigits value = ‘’2’’> 2 nombre apres la virgule.

Les types complexes : le type complexe contient d’autres éléments et des attributs et il est déclaré par l’élément complexType, la balise séquence permet d’introduire une liste d’éléments qu’on peut contrôler leur occurrence par les attributs minOccurs et maxOccurs.

Les attributs sont introduits par la balise attribute, qui possède :

* 1. Name = ‘’’’ 🡪 pour specifier le nom.
  2. Type = ‘’’’ 🡪 pour specifier le type.
  3. Use = ‘’’’ 🡪required, optional, prohibited.
  4. Fixed, default = ‘’defaultValue’’.

Exemples la liste contient 0 ou plusieurs éléments, et a une attribut année qui fixée en 2018. 