# XML

## Historique

Dérive de SGML (Standard Generalized Markup Language)

## Balises

- Par paires ou uniques

- Contenu :

* texte
* autre balise : arborescence

- Le chevauchement est interdit

- Règles de nommage :

* Les noms peuvent contenir des lettres, des chiffres ou des caractères spéciaux.
* Les noms ne peuvent pas débuter par un nombre ou un caractère de ponctuation.
* Les noms ne peuvent pas commencer par les lettres XML (quelle que soit la casse).
* Les noms ne peuvent pas contenir d'espaces.
* On évitera les caractères - , ; . < et > qui peuvent être mal interprétés dans vos programmes.

- Attributs :

* Mêmes règles de nommage
* Délimités par des " " ou des ' '
* Un attribut doit être unique pour une balise donnée

- commentaires :

<!-- Commentaire -->

<balise attribut="valeur">

contenu

</balise>

- Prologue :

<?xml version="1.0" encoding="UTF-8"?>

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

Standalone = yes : interdit de faire référence à des DTD externes (standalone tjrs en dernier)

- Corps

Balise racine unique

# DTD

Règles de base : Document **BIEN FORME**

Définition de données : Document **VALIDE**

DTD = Document Type Definition

* DTD internes : dans le fichier XML

<!DOCTYPE racine [ ]>

* DTD externes : fichier séparé (.dtd)

<!DOCTYPE racine PUBLIC "identifiant" "url">

<!DOCTYPE racine SYSTEM "uri">

## Eléments

**ELEMENT balise contenu**

contenu :

* EMPTY
* ANY *(déconseillé)*
* #PCDATA : valeur simple
* Balises :
  + **,** : séquence : elt1, elt2, elt3 ...
  + **|**  : liste de choix (exclusif) : elt1 | elt2 | elt3 ...
  + **?** : optionnel (0..1) : elt1, elt2Optionnel?, elt3...
  + **\*** : optionnel (0..\*) : elt1, elt2Optionnel\*, elt3...
  + **+** : répété (1..\*) : elt1, elt2Répété+, elt3...

<!ELEMENT formateurs (personne+)>

<!ELEMENT personne (nom, prenom, email, adresse)>

<!ELEMENT nom (#PCDATA)>

## Attributs

**ATTLIST balise attribut type mode**

type :

* **CDATA** : texte non parsé (libre)
* **|** : liste de valeurs : (val1 | val2 |val3)
* **ID** : identifiant unique
* **IDREF** : référence à un ID (clé étrangère)

mode :

* **#REQUIRED** : obligatoire
* **#IMPLIED** : optionnel
* **"valeur"** : valeur par défaut
* **#FIXED "valeur"** : constante (présente ou par défaut)

## Entités

* générales
* paramètres
* externes

**Entités générales**

Dans le document XML

ENTITY nom "valeur"

usage : **&entité;**

**Entités paramétrées**

Dans la DTD

ENTITY % nom "valeur"

**Entités externes**

ENTITY nom SYSTEM "uri"

# Schemas XML

fichiers .xsd (format xml)

Corps :

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

...

</xsd:schema>

Référencement dans un fichier XML :

<racine xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="monSchema.xsd">

...

</racine>

## Eléments Simples

<xsd:element name="nom" type="xsd:string"/>

Valeur par défaut : default="valeur"

Constante : fixed="valeur"

## Attributs

<xsd:attribute name="sexe" type="xsd:string"/>

Valeur par défaut : default="valeur"

Constante : fixed="valeur"

Obligatoire : use="required"

## Principaux types simples

préfixés par le namespace du schema (généralement "xsd:")

* Chaines de caractères
  + **string**
  + **language**
  + **ID** (attributs seulement)
  + **IDREF** (attributs seulement)
  + **IDREFS** (attributs seulement)
* Numériques
  + **integer**, **long**, **int**, **short**, **byte**
  + **float**, **double**, **decimal**
  + positiveX / nonPositiveX / negativeX / nonNegativeX / unsignedX
* Dates
  + **date** : YYYY-MM-DD
  + **time** : hh:mm:ss
  + **dateTime** : YYYY-MM-DDT hh:mm:ss
  + **duration** : PxYxMxDTxHxMxS
  + **gYear**, **gMonth** (--), **gDay**(---), ...
* Autres
  + **boolean** : true (1) / false (0)
  + **anyURI** : uri
  + **hexBinary** : hexdécimal

## Eléments complexes

Déclaration :

<xsd:element name="personne">

<xsd:complexType>

...

</xsd:complexType>

</xsd:element>

Contenus :

* simples : attributs et contenu texte (pas d'éléments imbriqués)

<xsd:simpleContent>

<xsd:extension base="xsd:string">

<xsd:attribute name="type" type="xsd:string"/>

<xsd:attribute name="privee" type="xsd:boolean" default="false"/>

</xsd:extension>

</xsd:simpleContent>

* xml : attributs et éléments imbriqués (pas de texte)

<xsd:complexType>

<xsd:sequence minOccurs="n" maxOccurs="n" > (tous, dans l'ordre) / all (ordre indifférent) / choice (ou exclusif)

<xsd:element name="nom" type="xsd:string"></xsd:element>

<xsd:element name="prenom" type="xsd:string"></xsd:element>

</xsd:sequence>

<xsd:attribute name="sexe" type="xsd:string" ></xsd:attribute>

</xsd:complexType>

* mixtes : attributs, éléments imbriqués et contenu texte

<xsd:complexType mixed="true">

...

</xsd:complexType>

Choix :

<xsd:element name="personne">

<xsd:complexType>

<xsd:choice>

<xsd:element name="nom" type="xsd:string"></xsd:element>

<xsd:element name="prenom" type="xsd:string"></xsd:element>

</xsd:choice>

</xsd:complexType>

</xsd:element>

## Référence à un élément :

<xsd:element ref="personne">...

## Héritage

**Types simples : restriction**

|  |  |
| --- | --- |
| Nom de la restriction | Description |
| minExclusive | permet de définir une valeur minimale exclusive |
| minInclusive | permet de définir une valeur minimale inclusive |
| maxExclusive | permet de définir une valeur maximale exclusive |
| maxInclusive | permet de définir une valeur maximale inclusive |
| totalDigits | permet de définir le nombre exact de chiffres qui composent un nombre |
| fractionDigits | permet de définir le nombre de chiffres autorisés après la virgule |
| Length | permet de définir le nombre exact de caractères d'une chaîne |
| minLength | permet de définir le nombre minimum de caractères d'une chaîne |
| maxLength | permet de définir le nombre maximum de caractères d'une chaîne |
| enumeration | permet d'énumérer la liste des valeurs possibles |
| whiteSpace | permet de déterminer le comportement à adopter avec les espaces |
| pattern | permet de définir des expressions régulières |

<xsd:attribute name="sexe">

<xsd:simpleType>

<xsd:restriction base="xsd:string">

<xsd:enumeration value="masculin"/>

<xsd:enumeration value="feminin"/>

</xsd:restriction>

</xsd:simpleType>

</xsd:attribute>

**Types complexes : extension**

<xsd:complexType name="stagiaire">

<xsd:complexContent>

<xsd:extension base="personne">

<xsd:sequence>

<xsd:element name="promo" type="xsd:string" nillable="false"></xsd:element>

</xsd:sequence>

<xsd:attribute name="delegue" type="xsd:boolean" default="false"></xsd:attribute>

<xsd:attribute name="prof" type="xsd:IDREF"></xsd:attribute>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

# XPath

Chemins relatifs / absolus

syntaxe : axe::nœud[prédicat]

## Axe

|  |  |
| --- | --- |
| Nom de l'axe | Description |
| ancestor | Ancêtres du nœud courant |
| ancestor-or-self | Nœud courant et ses ancêtres |
| attribute | Attributs du nœud courant |
| child | Enfants du nœud courant *(par défaut)* |
| descendant | Descendants du nœud courant |
| descendant-or-self | Nœud courant et ses descendants |
| following | Nœuds suivant le nœud courant |
| following-sibling | Frères suivants du nœud courant |
| parent | Père du nœud courant |
| preceding | Nœuds précédant le nœud courant |
| preceding-sibling | Frères précédents du nœud courant |
| self | Nœud courant |

## Nœud

|  |  |
| --- | --- |
| Nom | Description |
| nom du nœud | Nom du nœud |
| \* | Tous les nœuds |
| node() | Tous les types de nœuds (éléments, commentaires, attributs, etc.) |
| text() | Nœuds de type texte |
| comment() | Nœuds de type commentaire |

## Prédicats

|  |  |
| --- | --- |
| Nom du prédicat | Description |
| attribute | Recherche par attribut |
| count() | Nombre de nœuds |
| last() | Dernier nœud d'une liste |
| position() | Recherche par position d'un nœud |

## Fonctions XPath

sum()

concat()

…