

# XSD

## (XML Schema Definition)

## Définition:

Comme pour les DTD , le XSD permet de définir un Schéma XML et décrire la structure d'un document XML.

Le but d'un schéma XML est de définir les blocs de construction légaux d'un document XML:

- les éléments et attributs pouvant apparaître dans un document
- le nombre (et la commande) des éléments enfants
- types de données pour les éléments et les attributs
- valeurs par défaut et fixes pour les éléments et les attributs

Un fichier dans lequel est écrit un Schéma XML porte l'extension ".xsd"

## La structure d'un document XSD:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
</xsd:schema>
```

La présence d'un **élément racine**, c'est-à-dire la présence d'une balise qui contient toutes les autres. Mais, contrairement à un fichier XML, son nom nous est imposé.

L'élément racine est `<xsd:schema />`.

la présence de l'attribut `xmlns:xsd`. `xmlns` nous permet de déclarer un **espace de noms**.

A travers la déclaration de cet **espace de noms**, tous les éléments doivent commencer par **xsd**

### Exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<liste_employes>
  <employe>
    <matricule>23432</matricule>
    <nom>HUTIN</nom>
    <prenom>Sylvie</prenom>
    <adresse>16, rue des Passages – 91150 ETAMPES</adresse>
  </employe>
  <employe>
    <matricule>56435</matricule>
    <nom>JOLUN</nom>
    <prenom>Françoise</prenom>
    <adresse>112, avenue des Carreaux – 91580 ETRECHY</adresse>
  </employe>
</liste_employes>
```

Définit le fichier xsd correspondant au fichier xml :

```
<xsi:schemaLocation="chemin_vers_fichier.xsd">
```

## Déclaration d'un élément simple:

Un **élément simple** est un élément qui ne contient qu'une valeur. Il ne contient pas d'autres éléments, dont le type est dit **simple**.

```
<nom>Rachid</nom>
```

Il ne contient ni attribut ni aucun autre élément

```
<xsd:element name="mon_nom" type="xsd:mon_type" />
```

```
<nom>DUPONT</nom>
```

```
<prenom>Robert</prenom>
```

```
<age>38</age>
```

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

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

```
<xsd:element name="age" type="xsd:int" />
```

Le schéma permet d'indiquer le type des éléments du document :

<b>string</b>	Texte
<b>data</b>	Dates
<b>integer</b>	Nombre entier (1, 2, 3...)
<b>decimal</b>	Nombre réels (1.23, 0.17...)

## Valeur par défaut et valeur inchangeable

### Valeur par défaut

il est tout à fait possible d'indiquer dans les Schémas XML qu'un élément a une valeur par défaut

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

### Valeur constante

il est également possible d'imposer une valeur. Cette valeur inchangeable est appelée **constante**.

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

### Déclaration des attributs

```
<xsd:attribut name="mon_nom" type="xsd:mon_type" />
```

Exemple:

```
<personne sexe="masculin">Robert Rachid</personne>
```

```
<xsd:attribut name="sexe" type="xsd:string" />
```

## Valeur par défaut, obligatoire et interchangeable

### Valeur par défaut

```
<xsd:attribut name="num" type="xsd:string" default="A" />
```

### Valeur constante

```
<xsd:attribut name="tel" type="xsd:string" fixed="oooooooo" />
```

### Attribut obligatoire

```
<xsd:attribut name="prenom" type="xsd:string" use="required" />
```

Exemple:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<liste_employes>
  <employe>
    <matricule>23432</matricule>
    <nom>HUTIN</nom>
    <prenom>Sylvie</prenom>
    <adresse>16, rue des Passages - 91150 ETAMPES</adresse>
  </employe>
  <employe>
    <matricule>56435</matricule>
    <nom>JOLUN</nom>
    <prenom>Françoise</prenom>
    <adresse>112, avenue des Carreaux - 91580 ETRECHY</adresse>
  </employe>
</liste_employes>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="liste_employe">
    <xsd:complexType>
      <xsd:sequence maxOccurs="1" minOccurs="1">
        <xsd:element name="employe">
          <xsd:complexType>
            <xsd:sequence maxOccurs="unbounded" minOccurs="1">
              <xsd:element name="matricule" type="xsd:string"/>
              <xsd:element name="nom" type="xsd:string"/>
              <xsd:element name="prenom" type="xsd:string"/>
              <xsd:element name="adresse" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```



## Les types complexes:

Un **élément complexe** est un élément qui contient d'autres éléments ou des attributs.

<!-- la balise personne contient d'autres balises => élément complexe -->

```
<personne>
  <!-- la balise nom est un élément simple -->
  <nom>ROBERT</nom>
  <!-- la balise prenom est un élément simple -->
  <prenom>Axel</prenom>
</personne>
```

<!-- la balise personne possède un attribut => élément complexe -->

```
<personne sexe="feminin">Axel ROBERT</personne>
```

## Déclaration d'un élément complexe

```
<xsd:element name="personne">
  <xsd:complexType>
    <!-- contenu ici -->
  <xsd:complexType>
</xsd:element>
```

## Élément complexe ne contenant pas d'attributs:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<courriers>
  <lettre>
    <destinataire>Mme CORY</destinataire>
    <date>13/01/07</date>
    <contenu>Congé maternité</contenu>
  </lettre>
  <lettre>
    <destinataire>Mme TILLY</destinataire>
    <date>13/01/07</date>
    <contenu>Avancement</contenu>
  </lettre>
</courriers>
```

```
<xsd:element name="courriers">
  <xsd:complexType>
    <xsd:sequence maxOccurs="1" minOccurs="1">
```

```
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

le schéma permet de préciser le nombre d'occurrences (nombre de fois ou sont répétés les éléments).

<b>minOccurs=</b>	"0" ou "1"
<b>maxOccurs=</b>	"1" ou "unbounded" (illimité)

Exemple:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<courrier>
  <personne>
    <nom>Martin</nom>
    <prenom>Marie-Frédérique</prenom>
    <numero>123</numero>
    <rue1>Bâtiment B1</rue1>
    <rue2>allée des fleurs</rue2>
    <cp>91150</cp>
    <ville>Etampes</ville>
    <entete>Votre départ en retraite</entete>
    <objet>Très chère Madame, nous sommes heureux de vous annoncer votre départ
en retraite pour le 15/03/06.</objet>
  </personne>
  <personne>
    <nom>Moulin</nom>
    <prenom>Marie</prenom>
    <numero>13</numero>
    <rue1>allée des peupliers</rue1>
    <cp>91510</cp>
    <ville>Lardy</ville>
    <entete>Votre changement d'indice</entete>
    <objet>Très chère Madame, nous sommes heureux de vous annoncer votre
changement d'indice (n°454) à compter du 15/03/06.</objet>
  </personne>
</courrier>
```

## Réponse:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="COURRIER">
    <xs:complexType>
      <xs:sequence maxOccurs="1" minOccurs="1">
        <xs:element name="PERSONNE">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded" minOccurs="1">
              <xs:element maxOccurs="1" minOccurs="1" name="NOM" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="PRENOM" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="NUMERO" type="xs:integer"/>
              <xs:element maxOccurs="1" minOccurs="1" name="RUE1" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="0" name="RUE2" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="CP" type="xs:integer"/>
              <xs:element maxOccurs="1" minOccurs="1" name="VILLE" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="ENTETE" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="OBJET" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<embauche>
  <type emploi>
    <emploi>assistant commercial</emploi>
    <anciennete>3 ans</anciennete>
    <qualites>
      <critere>
        <intitule>Mobilité</intitule>
        <importance>5</importance>
      </critere>
      <critere>
        <intitule>Anglais (lu)</intitule>
        <importance>3</importance>
      </critere>
    </qualites>
  </type emploi>
  <type emploi>
    <emploi>assistant comptable</emploi>
    <anciennete>5 ans</anciennete>
    <qualites>
      <critere>
        <intitule>Rigueur</intitule>
        <importance>5</importance>
      </critere>
      <critere>
        <intitule>Anglais (lu)</intitule>
        <importance>4</importance>
      </critere>
    </qualites>
  </type_emploi>
</embauche>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="EMBAUCHE">
    <xs:complexType>
      <xs:sequence maxOccurs="1" minOccurs="1">
        <xs:element name="TYPE_EMPLOI">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded" minOccurs="1">
              <xs:element maxOccurs="1" minOccurs="1" name="EMPLOI" type="xs:string"/>
              <xs:element maxOccurs="1" minOccurs="1" name="ANCIENNETE" type="xs:integer"/>
              <xs:element name="QUALITES"/>
            <xs:complexType>
              <xs:sequence maxOccurs="1" minOccurs="1">
                <xs:element name="CRITERE"/>
              <xs:complexType>
                <xs:sequence maxOccurs="unbounded" minOccurs="1">
                  <xs:element maxOccurs="1" minOccurs="1" name="INTITULE" type="xs:string"/>
                  <xs:element maxOccurs="1" minOccurs="1" name="IMPORTANCE" type="xs:integer"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

```
<product prodid="1345" />
```

```
<xs:element name="product">  
  <xs:complexType>  
    <xs:attribute name="prodid" type="xs:positiveInteger"/>  
  </xs:complexType>  
</xs:element>
```





