

XSD

(XML Schema Definition)

## C'est quoi un document XSD:

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="personnes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="personne">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="nom"/>
              <xs:element type="xs:string" name="prenom"/>
            </xs:sequence>
            <xs:attribute type="xs:string" name="sexe"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## **Pourquoi le XSD:**

- **Définir Les éléments XML et leurs types (simple ou complexe).**
- **Définir les occurrences des éléments XML.**
- **Définir Les éléments XML et leurs types (simple ou complexe).**
- **Définir les Contraintes des éléments XML et leurs Valeurs.**

## La structure d'un document XSD.

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

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

Le prologue:

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

L'espace de nom : `xmlns:xsd`

## Association d'un XSD avec XML

```
<repertoire  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="biblio10.xsd">  
  .....  
</reptrtoire>
```

### L'espace de noms

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

### La location

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

## Déclaration d'un élément simple:

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

Exemple:

```
<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" />
```

## Les types d'un élément XSD:

|                |                              |
|----------------|------------------------------|
| <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...) |

xs:string

xs:decimal

xs:integer

xs:boolean

xs:date

xs:time



## Valeur par défaut et valeur inchangeable

### Valeur par défaut

il est tout à fait possible d'indiquer dans les Schémas XML

```
<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 d'un élément complexe ou séquence d'éléments:

Exemple:

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

Déclaration d'un élément complexe et son type:

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

## La séquence et Les occurrences d'un élément complexe:

La séquence définit les éléments XML contenu dans un autre élément

XML: `<?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>`

Chaque séquence définit le nombre d'occurrence d'un élément XML dans le document XML.

```
<?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">

|
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## L'occurrence min et l'occurrence max

`minOccurs=`

"0" ou "1"

`maxOccurs=`

"1" ou "unbounded" (illimité)

Dans un élément complexe le nombre d'occurrence d'un élément simple est par défaut :

## 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>
```

## Déclaration des attributs dans un Document XSD:

Exemple:

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

```
<xsd:element name= personnes  
<xsd: complexeType>
```

```
</xsd: complexeType>  
</xsd:element>
```



**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="00000" />
```

## **Attribut obligatoire**

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

Exemple:

```
<nom sexe=" M"> Rachid </nom>
```

```
</xs:schema>
```

```
<xs:element name="nom">
```

```
<xs:complexType>
```

```
<xs:attribute type="xs:string" name="sexe"/>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
</xs:schema>
```

Pour un **élément Complexe** qui contient un attribut on le déclare comme suit:

```
<?xml version="1.0"?>  
  
  <personne sexe="f">  
  
    <nom>a</nom>  
  
    <prenom>a</prenom>  
  
  </personne>
```

le code XSD pour valider le XML suivant:

Réponse: <https://www.freeformatter.com/xsd-generator.html>

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="personnes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="personne">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="nom"/>
              <xs:element type="xs:string" name="prenom"/>
            </xs:sequence>
            <xs:attribute type="xs:string" name="sexe"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A blue oval with a dark blue border, centered on the page.

Examen 2018 TH

Correction

## Les restrictions en XSD (L'héritage):

L'élément Restriction est utilisé pour définir les valeurs acceptées qu'un élément XML peut prendre.

### Restriction sur la valeur.

L'âge doit être comprises entre 0 et 100

```
<xs:element name = "age">
  <xs:simpleType>
    <xs:restriction base = "xs:integer">
      <xs:minInclusive value = "0"/>
      <xs:maxInclusive value = "100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

## Restriction sur un ensemble de valeurs.

La mention ne doivent être que A, B ou C.

```
<xs:element name = "mention">
  <xs:simpleType>
    <xs:restriction base = "xs:string">
      <xs:enumeration value = "A"/>
      <xs:enumeration value = "B"/>
      <xs:enumeration value = "C"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

## Restriction utilisant une expression régulière.

le prénom doit être en alphabet uniquement

```
<xs:element name = "firstname">  
  <xs:simpleType>  
    <xs:restriction base = "xs:string">  
      <xs:pattern value = "[a-z]"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```



## Exemple:

```
<?xml version="1.0" ?>
<xs:schema >
  <!-- Les types -->
  <xs:simpleType name="nompersonne">
    <xs:restriction base="xs:string">
      <xs:minLength value="5" />
      <xs:maxLength value="25" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="idstring">
    <xs:restriction base="xs:string">
      <xs:maxLength value="8" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="adresseweb">
    <xs:restriction base="xs:string">
      <xs:pattern value="http://.+" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="nomfichier">
    <xs:restriction base="xs:string">
      <xs:pattern value="\w+\.\w{3}" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="montypedate">
    <xs:restriction base="xs:string">
      <xs:pattern value="\d\d? \w+ \d{4}" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="mot">
    <xs:restriction base="xs:string">
      <xs:pattern value="\S+" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Les éléments -->
  <xs:element name="nom" type="nompersonne" />
  <xs:element name="id" type="idstring" />
  <xs:element name="adresse" type="adresseweb" />
  <xs:element name="fichier" type="nomfichier" />
  <xs:element name="date" type="montypedate" />
  <xs:element name="mot" type="mot" />
</xs:schema>
```







