

1. Elementos simples
2. Atributos en XSD
3. Restricciones en XSD
4. Elementos complejos

XML Schema (Parte II)

Elementos simples

Definición

```
<xs:element name="xxx" type="yyy"/>
```



Nombre del elemento

Definición

```
<xs:element name="xxx" type="yyy"/>
```

Tipo de dato:
xs:string, xs:decimal, xs:integer,
xs:boolean, xs:date, xs:time

Ejemplo

XSD:

```
<xs:element name="lastname" type="xs:string"/>  
<xs:element name="age" type="xs:integer"/>  
<xs:element name="dateborn" type="xs:date"/>
```



Valid XML:

```
<lastname>Refsnes</lastname>  
<age>36</age>  
<dateborn>1970-03-27</dateborn>
```



Ejemplo

XSD:

```
<xs:element name="lastname" type="xs:string"/>  
<xs:element name="age" type="xs:integer"/>  
<xs:element name="dateborn" type="xs:date"/>
```



Valid XML:

```
<lastname>Refsnes</lastname>  
<age>36</age>  
<dateborn>1970-03-27</dateborn>
```



Ejemplo

XSD:

```
<xs:element name="lastname" type="xs:string"/>  
<xs:element name="age" type="xs:integer"/>  
<xs:element name="dateborn" type="xs:date"/>
```



Valid XML:

```
<lastname>Refsnes</lastname>  
<age>36</age>  
<dateborn>1970-03-27</dateborn>
```



Valor por defecto para un elemento

```
<xs:element name="color" type="xs:string" default="red"/>>
```

Valor por defecto para el elemento "color"
es "red"

Valor fijo para un elemento

```
<xs:element name="color" type="xs:string" fixed="red"/>>
```

Se asigna el valor "red" para el elemento "color" y no se podrá cambiar

Atributos en XSD

Definición

```
<xs:attribute name="xxx" type="yyy"/>
```



Nombre del elemento

Definición

```
<xs:attribute name="xxx" type="yyy"/>
```

Tipo de dato:
xs:string, xs:decimal, xs:integer,
xs:boolean, xs:date, xs:time

Restricciones en XSD

Restricciones en valores

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

El elemento "age" no puede tener un valor menor que 0 ni mayor que 120

Restricciones a un conjunto de valores

```
<xs:element name="car">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Audi"/>
      <xs:enumeration value="Golf"/>
      <xs:enumeration value="BMW"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

El elemento "car" se restringe a los valores Audi, Golf y BMW

Restricciones a una serie de valores

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

El elemento "letter" admite una letra minúscula entre la a y la z

Restricciones de longitud

```
<xs:element name="password">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:length value="8"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

El elemento "password" se restringe a una longitud igual a 8 caracteres



https://www.w3schools.com/xml/schema_facets.asp

Elementos complejos

Opción 1

```
<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

El elemento tiene sub-elementos

Opción 2

```
<xs:element name="employee" type="personinfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```



Referencia al tipo de elemento

Elemento vacío (ejemplo)

`<product pid="1345"/>`

Elemento con un atributo

Elemento vacío (definición)

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



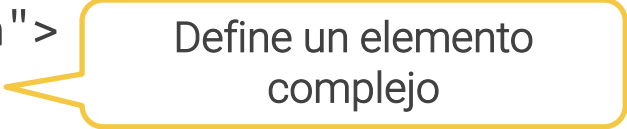
Define un elemento complejo

Elemento con sub-elementos (ejemplo)

```
<employee>  
  <firstname>John</firstname>  
  <lastname>Smith</lastname>  
</employee>
```

Elemento con sub-elementos (definición)

```
<xs:element name="person">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="firstname" type="xs:string"/>  
      <xs:element name="lastname" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```



Define un elemento complejo

Elemento con atributos (ejemplo)

```
<food type="dessert">Ice cream</food>
```

Elemento con atributos (definición)

```
<xs:element name="food">  
  <xs:complexType>  
    <xs:simpleContent>  
      <xs:extension base="xs:string">  
        <xs:attribute name="type" type="xs:string" />  
      </xs:extension>  
    </xs:simpleContent>  
  </xs:complexType>  
</xs:element>
```

Define un elemento complejo

Define el atributo

Elemento con contenido mixto (ejemplo)


`<letter>`
Dear Mr. `<name>John Smith</name>`.
Your order `<orderid>1032</orderid>`
will be shipped on `<shipdate>2001-07-13</shipdate>`.
`</letter>`

Elementos

Textos

Elemento con contenido mixto (definición)

```
<xs:element name="letter">  
  <xs:complexType mixed="true">  
    <xs:sequence>  
      <xs:element name="name" type="xs:string"/>  
      <xs:element name="orderid" type="xs:positiveInteger"/>  
      <xs:element name="shipdate" type="xs:date"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```



Elemento mixto

“No es que sea muy listo, es que me quedo durante más tiempo trabajando en los problemas”

ALBERT EINSTEIN

