**XSD （xml Schema Definition）**

XSD（xml模板定义）用来验证XML内容的有效性，使用xsd我们可以实现快速地编写xml文件，并且保证xml文件的正确性和有效性。具体作用如下：

1．定义一个Xml文档中都有什么元素

2．定义一个Xml文档中都会有什么属性

3．定义某个节点的都有什么样的子节点，可以有多少个子节点，子节点出现的 顺序

4．定义元素或者属性的数据类型

5．定义元素或者属性的默认值或者固定值

* **Xsd文件本质也是一个xml文件，它的根元素如下：**
* <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"（接下来以这个配置为例进行解释） 表示数据类型等定义来自w3，如果使用到w3中的标签，就必须加上前缀“xs:”。也可省略“xs”，比如定义为<schema xmlns:xs="http://www.w3.org/2001/XMLSchema"，这样，使用w3标签的时候就可以直接使用，不用在每个标签之前添加“xs：”。
* targetNamespace="http://www.w3schools.com" 表示文档中要定义的元素来自什么命名空间
* xmlns="http://www.w3schools.com"表示此文档的默认命名空间是什么
* elementFormDefault="qualified"> 表示要求xml文档的每一个元素都要有命名空间指定

……定义主体部分……

* </xs:schema>
* **使用element定义一个元素**
* <xs:element  此处表示要定义一个元素
* name=”color” 表示要定义元素的名称
* type=”xs:string”  表示要定义元素的数据类型
* default=”red” 表示定义元素的默认值
* fixed=”red”/> 表示要定义元素的固定值，此元素只可以取“red”值（如果设定了fixed，在编写xml的时候就只能给color元素赋red值）

以上定义了一个简单元素，元素实例：<color>red</color>

* 使用attribute定义属性
* <xs:attribute
* name=”birthday” 表示要定义属性的名字
* type=”xs:date” 表示要定义属性的数据类型
* default=”2001-01-11” 表示要定义属性的默认值
* fixed=”2001-01-11” 表示要定义属性的固定值
* use=”required”/> 表示此属性是否是为必填项，即如果不指定就不符合Schema，默认没有use=”required”属性表示属性可有可无
* 最大值最小值限制
* <xs:element name="age">
* <xs:simpleType>
* <xs:restriction base="xs:integer">
* <xs:minInclusive value="0"/> 大于等于0，<xs: minExclusive>表示最小值但是不包括指定值
* <xs:maxInclusive value="120"/> 小于等于120，<xs: maxExclusive>
* </xs:restriction>
* </xs:simpleType>
* </xs:element>
* 枚举限制，指只能在指定的几个值中取值
* <xs:element name="car" type="carType"/>

<xs:simpleType name="carType">

   <xs:restriction base="xs:string">

     <xs:enumeration value="Audi"/>

     <xs:enumeration value="Golf"/>

     <xs:enumeration value="BMW"/>

   </xs:restriction>

</xs:simpleType>

* 字符串长度的限制

<xs:element name="password">

<xs:simpleType>

  <xs:restriction base="xs:string">

    <xs:length value="8"/>

  </xs:restriction>

</xs:simpleType>

</xs:element>

* 此外还有模式限制，指字符串的格式必须满足制定的匹配模式
* 复杂类型是指定义元素中包含属性或者子元素的类型
* 定义只包含子元素的复杂类型

<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>

* 定义只包含属性的复杂类型

<xs:element name="product" type="prodtype"/>

<xs:complexType name="prodtype">

  <xs:attribute name="prodid" type="xs:positiveInteger"/>

</xs:complexType>

* 定义只包含内容的复杂类型

<xs:element name="shoesize" type="shoetype"/>

<xs:complexType name="shoetype">

  <xs:simpleContent>

    <xs:extension base="xs:integer">

      <xs:attribute name="country" type="xs:string" />

    </xs:extension>

  </xs:simpleContent>

</xs:complexType>

* 定义包含内容和子元素混合的复杂类型

<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>