2) Anonymous Data Type

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

    <xsd:element name="grade">
        <xsd:simpleType>
            <xsd:restriction base="xsd:positiveInteger">
                <xsd:minInclusive value="4"/>
                <xsd:maxInclusive value="10"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>

</xsd:schema>
```

Benefits of Named Data Type

- If you want re-use your datatype:

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

   <xsd:element name="grade" type="grade_type" />
   <xsd:element name="teachers_IQ" type="grade_type" />

    <xsd:simpleType name="grade_type">
        <xsd:restriction base="xsd:positiveInteger">
            <xsd:minInclusive value="4"/>
            <xsd:maxInclusive value="10"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

SimpleType: enumeration

- Alternative content

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

SimpleType: pattern

- Using REGEX:

```
<xsd:simpleType name="lowercase_char">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[a-z]"/>
  </xsd:restriction>
</xsd:simpleType>
```

REGEX Examples

```
<xs:pattern value="[A-Z][A-Z][A-Z]"/>
<xs:pattern value="[a-zA-Z][a-zA-Z][a-zA-Z]"/>
<xs:pattern value="[xyz]"/>
<xs:pattern value="[0-9][0-9][0-9][0-9][0-9]"/>
<xs:pattern value="([a-z])*"/>
<xs:pattern value="male|female"/>
<xs:pattern value="[a-zA-Z0-9]{8}"/>
```

Structure of the XML-file

- It's possible to define the structure of the XML-file using `complexType`
- If element A has child-elements, then element A's type is `complexType`

SimpleType vs. ComplexType

- **SimpleType**
  - **<grade>**7**</grade>**
  - Since `grade` **does not** hold other child – elements, grade's type is **simpleType**
- **ComplexType**
  - **<students>**<student>Jack</student>**</students>**
  - Since `student` **does hold** child – element(s), student's type is **complexType**

Example: XML - File

```
<?xml version="1.0"?>
<students>
   <firstname>Fernando</firstname>
  <lastname>Alonso</lastname>
</students>
```

Example: XSD – file
Named ComplexType

Use now complexType (vs. simpleType)

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

    <xsd:element name="students" type="students_type">

    <xsd:complexType name="students_type">
        <xsd:sequence>
            <xsd:element name="firstname" type="xsd:string"/>
            <xsd:element name="lastname"  type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

</xsd:schema>
```

Example: XSD – file
Anonymous ComplexType

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

    <xsd:element name="students">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="firstname" type="xsd:string"/>
                <xsd:element name="lastname"  type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

Example: ComplexType

```
<xsd:element name="employee" type="personinfo"/>
<xsd:element name="student"  type="personinfo"/>
<xsd:element name="member"   type="personinfo"/>

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

Deep Structure in XML - File

```
<?xml version="1.0"?>
<students>
   <student>
      <name>
          <firstname>Fernando</firstname>
      </name>
   </student>
</students>
```

Using Anonymous Data Type: The Horror!

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

    <xsd:element name="students">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="student">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="name">
                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element name="firstname" type="xsd:string"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

"There is an error in my schema, could you find it for me?"

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

<xsd:element name="students">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="student">
  <xsd:complexType>
   <xsd:sequence>
<xsd:element name="name">
 <xsd:complexType>
   <xsd:sequence>
 <xsd:element name="firstname" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
 </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>

</xsd:complexType>
</xsd:element>

</xsd:schema>
```

Use Named Datatypes! It's easier to find errors..

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

  <xsd:element name="students" type="students_type" />

  <xsd:complexType name="students_type">
    <xsd:sequence>
        <xsd:element name="student" type="student_type" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="student_type">
    <xsd:sequence>
        <xsd:element name="name" type="name_type" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="name_type">
    <xsd:sequence>
        <xsd:element name="firstname" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

Order of the elements

- **Sequence**: Elements appear in same order than in Schema
- **All**: Elements can appear in any order
- **Choice**: One element can appear from the choice-list

```
<xsd:element name="person">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element name="employee" type="employee"/>
      <xsd:element name="member" type="member"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Attribute

- XML
  - `<student **id="A1"**>...</student>`
- Schema

```
<xsd:element name="student"
 type="student_type" />

 <xsd:complexType name="student_type">
   <xsd:sequence>
        ...
   </xsd:sequence>
   <xsd:attribute name="id" type="xsd:ID"/>
 </xsd:complexType>
```

Empty Element with Attribute

- XML
  - `<student id="A1" />`
- Schema

```
<xsd:element name="student" type="student_type" />

 <xsd:complexType name="student_type">
    <xsd:attribute name="id" type="xsd:ID"/>
 </xsd:complexType>
```