

XML Schema

DTD a XML Schema

DTD jako opis dokumentu XML:

- pozwala stworzyć jedynie bardzo ogólny schemat dokumentu
- w standardzie DTD dostępne mamy jedynie rozróżnienie na elementy proste i złożone
- wewnątrz elementów prostych występuje tylko jeden typ danych, wewnątrz złożonych nie możemy dokładnie określić liczby wystąpień elementu

XML Schema – zastępuje DTD uzupełniając go o wiele nowych opcji:

- możemy dokładnie określać typy i wzorce zawartości elementów i także atrybutów
- wskazujemy też ilość i sekwencje występowania poszczególnych elementów

Budowa schematu XSD – plik z rozszerzeniem xsd.

Wskazanie przestrzeni nazw, z której zamierzamy korzystać w naszym pliku.

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    --- treść schematu ---

</xsd:schema>
```

[Odwołanie do przestrzeni nazw – xsd:http://www.w3.org/2001/XMLSchema -]

Schemat składa się, podobnie jak DTD, z deklaracji elementów i atrybutów:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="klient">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="imie" type="xsd:string" maxOccurs="2"/>
        <xsd:element name="nazwisko" type="xsd:string"/>
        <xsd:element name="data_urodzenia" type="xsd:date"/>
        <xsd:element name="pesel">
          <xsd:simpleType>
            <xsd:restriction base="xsd:integer">
              <xsd:pattern value="\d{11}"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="wiek">
          <xsd:simpleType>
            <xsd:restriction base="xsd:integer">
              <xsd:minInclusive value="0"/>
              <xsd:maxInclusive value="100"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
      </xsd:sequence>

      <xsd:attribute name="nr" type="xsd:integer"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

[przykładowy schemat – pogrubione przykładowe deklaracje elementu i atrybutu]

Dołączenie schematu do pliku XML

Odwołanie do schematu zawarte w pliku XML.

```
<?xml version="1.0" encoding="UTF-8"?>

<biblioteka xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="elementy.xsd">

    --- tresc ---

</biblioteka>
```

Typy elementów XSD

Definicje elementów prostych:

- zawierają tylko tekst (w określonym formacie)
- nie mogą zawierać innych elementów ani atrybutów

```
<xsd:element name="..." type="..." />
```

Niektóre wbudowane typy danych	
xsd:string	tekst
xsd:decimal	liczby dziesiętne
xsd:integer	liczby całkowite
xsd:Boolean	wartość prawda / fałsz
xsd:date	data
xsd:time	czas

Przykłady:

```
<xsd:element name="nazwa" type="xsd:string"/>
<xsd:element name="rok" type="xsd:integer"/>
<xsd:element name="data" type="xsd:date"/>
```

Wartości domyślne

```
<xsd:element name="..." type="..." default="..." />
```

Przykłady:

```
<xsd:element name="rok" type="xsd:integer" default="2003"/>  
<xsd:element name="typ" type="xsd:string" default="publiczna"/>
```

Wartości stałe

```
<xsd:element name="..." type="..." fixed="..." />
```

Przykłady:

```
<xsd:element name="rok" type="xsd:integer" fixed="2003"/>  
<xsd:element name="typ" type="xsd:string" fixed="publiczna"/>
```

Ograniczenia

Przedział wartości od 0 do 100.

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

Lista możliwych wartości.

```
<xsd:element name="miasto">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="Grodzisk"/>  
      <xsd:enumeration value="Milanówek"/>  
      <xsd:enumeration value="Brwinów"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

Ograniczenia długości

```
<xsd:simpleType>  
  <xsd:restriction base="xsd:string">  
    <xsd:length value="3"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

[długość wyrażenia ma wynosić dokładnie 3 znaki]

```
<xsd:simpleType>  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="3"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

[długość wyrażenia ma wynosić maksymalnie 3 znaki]

Wyrażenia regularne

```
<xsd:simpleType>  
  <xsd:restriction base="xsd:integer">  
    <xsd:pattern value="\d{4}"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

[wyrażenie składające się z 4 cyfr]

```
<xsd:simpleType>  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="[a-z][1-9]"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

[wyrażenie składające się z małej litery oraz cyfry (od 1 do 9)]

```
<xsd:simpleType>  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="A|B"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

[wyrażenie "A" lub "B"]

Wielokrotne wykorzystanie zdefiniowanego typu danych

```
<xsd:simpleType name="rok">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="[1|2][0-9][0-9][0-9]"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:element name="rok_budowy" type="rok">  
<xsd:element name="rok_oddania" type="rok">
```

[element rok_budowy oraz rok_oddania są typu "rok"]

Definicje elementów złożonych

Deklaracja typu złożonego

```
<xsd:element name="...">  
  <xsd:complexType>  
  
    --- elementy w odpowiedniej sekwencji ---  
  
  </xsd:complexType>  
</xsd:element>
```

Możliwe sekwencje elementów

```
<xsd:element name="biblioteka">  
  <xsd:complexType>  
  
    <xsd:sequence>  
  
      <xsd:element name="nazwa" type="xsd:string"/>  
      <xsd:element name="adres" type="xsd:string"/>  
      <xsd:element name="typ" type="xsd:string"/>  
  
    </xsd:sequence>  
  
  </xsd:complexType>  
</xsd:element>
```

[każdy z elementów zawarty w "sequence" jest obowiązkowy - zachowana kolejność]

```
<xsd:element name="biblioteka">
  <xsd:complexType>

    <xsd:all>

      <xsd:element name="nazwa" type="xsd:string"/>
      <xsd:element name="adres" type="xsd:string"/>
      <xsd:element name="typ" type="xsd:string"/>

    </xsd:all>

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

[każdy z elementów zawarty w "all" jest obowiązkowy - nie zachowana kolejność]

```
<xsd:element name="biblioteka">
  <xsd:complexType>

    <xsd:choice>

      <xsd:element name="nazwa" type="xsd:string"/>
      <xsd:element name="adres" type="xsd:string"/>
      <xsd:element name="typ" type="xsd:string"/>

    </xsd:choice >

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

[wystąpić musi dokładnie jeden element z "choice"]

```
<xsd:element name="biblioteka">
  <xsd:complexType>

    <xsd:sequence>
      <xsd:element name="nazwa" type="xsd:string"/>
      <xsd:element name="kontakt">
        <xsd:complexType>
          <xsd:choice>
            <xsd:element name="telefon" type="xsd:string"/>
            <xsd:element name="email" type="xsd:string"/>
          </xsd:choice>
        </xsd:complexType>
      <xsd:element name="typ" type="xsd:string"/>
    </xsd:sequence>

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

[kombinacja "sequence" i "choice"]

Liczba wystąpień elementu

```
<xsd:element name="imie" type="xsd:string" maxOccurs="2"/>
<xsd:element name="imie" type="xsd:string" maxOccurs="unbounded"/>
<xsd:element name="imie" type="xsd:string" minOccurs="1"/>
<xsd:element name="imie" type="xsd:string" minOccurs="0"/>
```

Grupy elementów

```
<xsd:group name="dane_osobowe">
  <xsd:sequence>
    <xsd:element name="imie" type="xsd:string"/>
    <xsd:element name="nazwisko" type="xsd:string"/>
    <xsd:element name="adres" type="xsd:string"/>
  </xsd:sequence>
</xsd:group>

<xsd:element name="kupujący">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="dane_osobowe"/>
      <xsd:element name="status" type="xsd:string"/>
      <xsd:element name="rabat" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="sprzedający">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="dane_osobowe"/>
      <xsd:element name="status" type="xsd:string"/>
      <xsd:element name="rabat" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```


Definicje atrybutów

```
<xsd:element name="faktura">
  <xsd:complexType>

    <xsd:attribute name="numer" type="xsd:string"/>
    <xsd:attribute name="typ" type="xsd:string"/>

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

```
<xsd:attribute name="numer" type="xsd:string" use="required"/>
<xsd:attribute name="numer" type="xsd:string" use="optional"/>
<xsd:attribute name="Numer" type="xsd:string" use="prohibited"/>
```

```
<xsd:attribute name="numer" type="xsd:string" use="required" default="01"/>
```

```
<xsd:attributeGroup name="naglowek">
  <xsd:attribute name="numer" type="xsd:string"/>
  <xsd:attribute name="typ" type="xsd:string"/>
</xsd:attributeGroup>

<xsd:element name="faktura">
  <xsd:complexType>

    <xsd:attributeGroup ref="naglowek"/>

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

Element pusty

```
<pozycja numer="12" />

<xsd:element name="pozycja">
  <xsd:complexType>
    <xsd:attribute name="numer" type="xsd:positiveInteger"/>
  </xsd:complexType>
</xsd:element>
```

Element o zawartości mieszanej

```
<osoba>
Wysokiej klasy specjalista
<imie>Kamil</imie> <nazwisko>Kolasiński</nazwisko>
Urodzony<data>1965-10-05</data>
</osoba>

<xsd:element name="osoba">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:element name="imie" type="xsd:string"/>
      <xsd:element name="nazwisko" type="xsd:string"/>
      <xsd:element name="data" type="xsd:date"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Element tekstowy z atrybutem

```
<firma typ="odbiorca">Miltia</firma>

<xsd:element name="firma">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="typ" type="xsd:string" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

Walidacja dokumentu XML i XSD

VS Code + dodatek XML od Red Hat

Netbeans

Plugin XML Tools do Notepad++

<http://www.xmlforasp.net/SchemaValidator.aspx>