

# **XML Vārdu Telpas un XML Schema**



# Plāns



- XML vārdu telpas
- XML dokumentu shēma un validācija
- Ieskats DTD dokumentu tipu definīcijā
- XML Schema
- XML Schema funkcijas un labumi
- XML Schema dokuments
- Vienkārša <element> deklarēšana
- Sarežģīta <element> deklarēšana
- Piemēri

# 1. XML vārdu telpas

- XML Namespaces (angl.)
- W3C rekomendācija

<http://www.w3.org/TR/REC-xml-names/>

- XML vārdu telpas ļauj izvairīties no dokumenta elementu vārdu konfliktiem
- Tiek izmantotas, lai atšķirtu elementus un atribūtus no dažādiem XML dokumentu veidiem (vārdnīcām), to apvienojot vienā dokumentā



## 2. Vārdu konflikts

- XML elementu vārdus definē izstrādātājs
- Vārdu konflikts var rasties apvienojot dažādus XML dokumentus
- Piemērā XML parsētājam nav starpības starp **<title>** dokumenta elementiem

```
<person>
  <name>
    <title>Sir</title>
    <first>John</first>
    <middle>Fitzgerald Johansen</middle>
    <last>Doe</last>
  </name>
  <position>Vice President of Marketing</position>
  <résumé>
    <html>
      <head>
        <title>Resume of John Doe</title>
      </head>
      <body>
        <h1>John Doe</h1>
        <p>John's a great guy, you know?</p>
      </body>
    </html>
  </résumé>
</person>
```

### 3. XML elementu prefikss (1)

XML elementu vārdu konflikta  
atrisināšanai katram elementam jābūt  
unikāli identificējamam

- Jāizmanto prefiksu elementa vārdā

**<prefix:elementName/>**



### 3. XML elementu prefikss (2)

```
<pers:person>
  <pers:name>
    <pers:title>Sir</pers:title>
    <pers:first>John</pers:first>
    <pers:middle>Fitzgerald Johansen</pers:middle>
    <pers:last>Doe</pers:last>
  </pers:name>
  <pers:position>Vice President of Marketing</pers:position>
  <pers:r  sum  >
    <xhtml:html>
      <xhtml:head>
        <xhtml:title>Resume of John Doe</xhtml:title>
      </xhtml:head>
      <xhtml:body>
        <xhtml:h1>John Doe</xhtml:h1>
        <xhtml:p>John's a great guy, you know?</xhtml:p>
      </xhtml:body>
    </xhtml:html>
  </pers:r  sum  >
</pers:person>
```

## 4. XML vārdu telpas (1)



- Lai izmantotu XML prefiksu, tam jādefinē vārdu telpa
- XML Namespaces rekomendācijas no W3C nosaka standartu vārda telpu deklarēšanai un elementa vai atribūta vārda telpas identificēšanai
- XML vārdu telpu izmantošanai dokumentā elementiem jāizmanto **kvalificētos** vārdus (QName)
- QName sastāv no divām daļām
  - Elementa lokāla daļa (<prefix:elementName>)
  - Vārda telpas prefikss (<prefix:elementName>)

## 4. XML vārdu telpas (2)

- XML vārdu telpa ir definēta izmantojot **xmlns** atribūtu elementa sākuma tagā
- XML vārdu telpas definēšanas sintakse  
**xmlns:prefix="URI"**
- Lai deklarētu vārdu telpu “**http://www.wiley.com/pers**” un saistītu **<person>** elementu ar to, ir jādefinē:  
**<pers:person**  
**xmlns:pers="http://www.wiley.com/pers"/>**



## 4. XML vārdu telpas (3)

- Prefiksu var izmantot jebkurš no **<pers:person>** elementa pectečiem, lai indicētu to piederību **http://www.wiley.com/pers** vārdu telpai

```
<pers:person xmlns:pers="http://www.wiley.com/pers">  
  <pers:name>  
    <pers:title>Sir</pers:title>  
  </pers:name>  
</pers:person>
```



## 5. XML vārdu telpa pēc noklusēšanas (1)

- Vārdu telpā pēc noklusēšanas nav jānorāda prefiksu visiem elementiem, kuri to izmanto

```
<person xmlns="http://www.wiley.com/pers">  
  <name>  
    <title>Sir</title>  
  </name>  
</person>
```



## 5. XML vārdu telpa pēc noklusēšanas (2)

- Elementam var definēt vairāk par vienu vārdu telpām, bet tikai viena varbūt pēc noklusēšanas

```
<person xmlns="http://www.wiley.com/pers"
        xmlns:xhtml="http://www.w3.org/1999/xhtml">
  <name />
  <xhtml:p>This is XHTML</xhtml:p>
</person>
```

## 5. XML vārdu telpa pēc noklusēšanas (3)

- XML vārdu telpu pēc noklusēšanas var definēt elementam un tā pēctečiem
- Piemērā vārdu telpa pēc noklusēšanas ir pārdefinēta **<p>** elementā uz **<http://www.w3.org/1999/xhtml>**.

```
<person xmlns="http://www.wiley.com/pers">
  <name />
  <p xmlns="http://www.w3.org/1999/xhtml">
    This is XHTML
  </p>
</person>
```

## 6. Atšķirīgas notācījas – vienāds rezultāts

```
<pers:person xmlns:pers="http://www.wiley.com/pers"
xmlns:xhtml="http://www.w3.org/1999/xhtml">
  <pers:name/>
  <xhtml:p>This is XHTML</xhtml:p>
</pers:person>
```

```
<person xmlns="http://www.wiley.com/pers"
xmlns:xhtml="http://www.w3.org/1999/xhtml">
  <name/>
  <xhtml:p>This is XHTML</xhtml:p>
</person>
```

```
<person xmlns="http://www.wiley.com/pers">
  <name/>
  <p xmlns="http://www.w3.org/1999/xhtml">This is XHTML</p>
</person>
```

## 7. XML vārdu telpu URI (1)

- Uniform Resource Identifier (angl.)
- Rakstzīmju virkne resursu identificēšanai
- URL (angl., Uniform Resource Locator)



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Response xmlns="http://tempuri/XMLSchema/v1"
  xmlns:ns2="http://ivis.eps.gov.lv/XMLSchemas/100017/fidavista/v1-2">
  <AccountStatementResponse>
    <ns2:FIDAVISTA>
      <ns2:Header>
        <ns2:Timestamp>20130312214257000</ns2:Timestamp>
        <ns2:From>PARXLV22</ns2:From>
      </ns2:Header>
      <ns2:Statement>
        <ns2:Period>
          <ns2:StartDate>2013-01-01</ns2:StartDate>
          <ns2:EndDate>2013-02-01</ns2:EndDate>
          <ns2:PrepDate>2013-09-27</ns2:PrepDate>
        </ns2:Period>
        <ns2:AccountSet>
          <ns2:IBAN>LV11PARX000123456789</ns2:IBAN>
        </ns2:AccountSet>
      </ns2:Statement>
    </ns2:FIDAVISTA>
  </AccountStatementResponse>
</Response>
```

## 7. XML vārdu telpu URI (2)

- URN (angl., Universal Resource Name).

```
<?xml version="1.0" encoding="UTF-8"?>
<CBICF:CB.ICF.BlkCdtTrf
  xmlns:CBICF="urn:CBICF:xsd:CB.ICF.BlkCdtTrf"

  <CBICF:FIToFICstmrCdtTrf
    xmlns="urn:iso:std:iso:20022:tech:xsd:sct:pacs.008.001.02.EKS">
    <GrpHdr>
      <MsgId>Payments001</MsgId>
      <CreDtTm>2010-06-30T08:31:00</CreDtTm>
      <NbOfTx>1</NbOfTx>
      <TtlIntrBkSttlmAmt Ccy="EUR">546.34</TtlIntrBkSttlmAmt>
      <IntrBkSttlmDt>2010-06-30</IntrBkSttlmDt>
      <SttlmInf>
```

## 7. XML vārdu telpu URI (3)

- W3C rekomendācija izmanto vārdu telpas **nosaukuma** terminu
- Bieži vārdu telpas URI definēšanai tiek izmantots URL HTTP shēmā (piem., <http://www.w3.org/1999/xhtml>), neskatoties uz XSD formālas saistības trūkumu ar HTTP protokolu
  - Izvēlēta URI apraksta resursu, kas atrodas XML vārdnīcas autora kontrolē. Piemēram, autora tīmekļa servera URL



# Piemērs: XML vārdu telpas XSLT

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <h2>My CD Collection</h2>
        <table border="1">
          <tr>
            <th style="text-align:left">Title</th>
            <th style="text-align:left">Artist</th>
          </tr>
          <xsl:for-each select="catalog/cd">
            <tr>
              <td>
                <xsl:value-of select="title" />
              </td>
              <td>
                <xsl:value-of select="artist" />
              </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
```

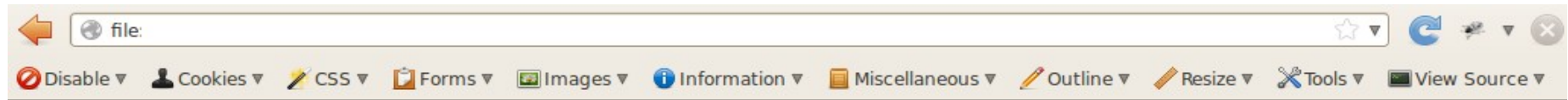
# Piemērs: ISO20022 maksājumu iniciēšana

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03"
  <CstmrCdtTrfInitn>
    <GrpHdr>
      <MsgId>87fbf20111125/1</MsgId>
      <CreDtTm>2011-11-25T11:16:58.696</CreDtTm>
      <NbOfTxes>3</NbOfTxes>
      <CtrlSum>2500</CtrlSum>
      <InitgPty>
        <Nm>AS XML</Nm>
      </InitgPty>
    </GrpHdr>
    <PmtInf>
      <PmtInfId>PMTID001</PmtInfId>
      <PmtMtd>TRF</PmtMtd>
      <BtchBookg>true</BtchBookg>
      <NbOfTxes>3</NbOfTxes>
      <PmtTpInf>
        <SvcLvl>
          <Cd>SEPA</Cd>
        </SvcLvl>
```

# Piemērs: XHTML 1.1 + MathML 2.0 + SVG 1.1

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:svg="http://www.w3.org/2000/svg">
  <body>
    <h2>MathML</h2>
    <p>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <mfrac>
          <mi>a</mi>
          <mi>b</mi>
        </mfrac>
      </math>
    </p>
    <h2>SVG</h2>
    <p>
      <svg:svg width="50px" height="50px">
        <svg:circle cx="25px" cy="25px" r="20px" fill="green" />
      </svg:svg>
    </p>
  </body>
</html>
```

# Piemērs: XHTML 1.1 + MathML 2.0 + SVG 1.1



**MathML**

$$\frac{a}{b}$$



**SVG**



# Praktiskais uzdevums



# XML Dokumentu Shēma

**DTD**  
**XML Schema**



# 1. XML shēma

- XML dokumenta kvalitātes kontroles rīks
- Apraksta XML dokumenta struktūru
- Ļauj pārļiecināties, ka XML dokumenta instance atbilst sintakses un struktūras noteikumu kopai
- XML shēmu var uztvert par programmu parsētājam kā lasīt dokumentu





## 2. XML shēmas valodas

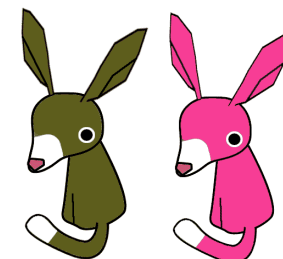
- DTD (angl. Document Type Definition)
  - vecākā XML dokumenta shēmas valoda
  - speciālā ne-XML sintakse



- **W3C XML Schema**
  - Pilnvērtīgs XML dokuments



- RELAX NG
- Schematron





# 3. XML dokumenta shēmas validācija

- Derīgs XML dokuments ir korekti noformēts XML dokuments, kurš arī atbilst XML dokumenta shēmas (DTD, XML Schema) noteikumiem



- **Validācijas process**
  1. Parsētājs ielāde shēmas noteikumus un deklarācijas
  2. Tiek konstruēts speciāla tipa validācijas parsētājs
  3. Parsētājs pieņem XML dokumenta instanci un ģenerē tas validācijas atskaiti

# 4. Ieskats DTD dokumentu tipu definīcijā

- (angl.) Document Type Definition
- Nosaka
  - atļautus XML dokumenta blokus
  - dokumenta struktūru, izmantojot sarakstu ar atļautiem dokumenta elementiem un atribūtiem
- DTD var specificēt XML dokumenta iekšā (internal) vai atsevišķajā failā (external)

# 4. Ieskats DTD dokumentu tipu definīcijā

- Iekšēja DTD piemērs
- `<!DOCTYPE root-element[element-declaration ]>`

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note[
    <!ELEMENT note (to,from,heading,body)>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body (#PCDATA)>
]>
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

# 5. XML Schema (1)

- “XML Schema” ir W3C tehnoloģija XML dokumenta shēmas aprakstam
- Viens no svarīgiem XML tehnoloģiju pamatiem
- Līdzīgi DTD XML Schema ļauj aprakstīt XML dokumenta struktūru
- XML Schema ir DTD alternatīva un pēctecis balstīts uz XML



## 5. XML Schema (2)

- Nodrošina datu atbilstību specifikācijai
- XML Schema valodu parasti sauc arī par XSD (angl. XML Schema Definition)
- XSD ir XML valoda
  - izmanto XML vārdu telpas
- XSD ir specifiskāka par DTD
  - izmanto datu tipus



# 6. XML Schema pamata funkcijas

- Definē kādi elementi var parādīties dokumentā
- Definē kādi elementu atribūti var parādīties dokumentā
- Definē kādu elementus var iekļaut citos elementos
- Definē elementu kārtību
- Nosaka iespējamo elementu skaitu
- Definē vai elements ir tukšs vai var iekļaut tekstu
- Definē datu tipus elementiem un atribūtiem
- Definē vērtības pēc noklusēšanas elementiem un atribūtiem

# 7. XML Schema labumi (1)

- XML Schema tiek izveidota izmantojot XML pamata sintaksi (atšķirībā no DTD speciālās sintakses)
- XML Schema pilnīgi atbalsta W3C Namespace rekomendācijas
- Atbalsta teksta elementu satura validāciju izmantojot iebūvētus un lietotāja definētus datu tipus



## 7. XML Schema labumi (2)

- Atļauj vienkārša veidā veidot sarežģītus un atkārtoti lietojamus satura modeļus (piem., ISO20022 ziņojumu standarts)
- Atbalsta programmēšanas valodu koncepciju modelēšanu (piem. mantošanas saistības no objekt-orientētās programmēšanas)





# 7. XML Schema labumi (3)

- ISO20022 ziņojumu formāta atkārtotā lietošana

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:lv8="urn:iso:std:iso:20022:tech:xsd:sct:pacs.008.001.02.EKS"
  xmlns:lv4="urn:iso:std:iso:20022:tech:xsd:sct:pacs.004.001.02.EKS"
  xmlns:NHICF="urn:NHICF:xsd:NH.ICF.BlkCdtTrf"
  targetNamespace="urn:NHICF:xsd:NH.ICF.BlkCdtTrf"
  elementFormDefault="qualified">

  <xs:import namespace="urn:iso:std:iso:20022:tech:xsd:sct:pacs.008.001.02.EKS"
    schemaLocation="pacs.008.001.02.EKS.xsd"/>
  <xs:import namespace="urn:iso:std:iso:20022:tech:xsd:sct:pacs.004.001.02.EKS"
    schemaLocation="pacs.004.001.02.EKS.xsd"/>
  <xs:element name="NH.ICF.BlkCdtTrf">

  <xs:annotation>
```

# 8. XML Schema vs. DTD (1)

- XML Schema ir paplašināma nākotnē
- XML Schema mehānism ir jaudīgāks
  - var atkārtoti lietot shēmas
  - var definēt speciālus datu tipus, atvasinot no iebūvētiem
  - var norādīt vairākas shēmas vienā dokumentā

**DTD vs XSD**

## 8. XML Schema vs. DTD (2)

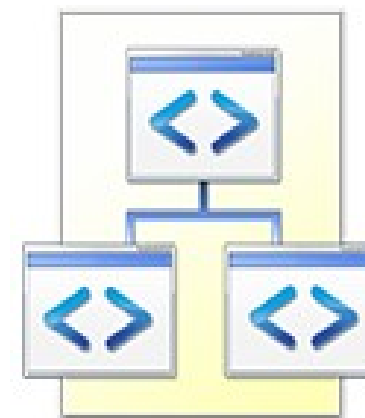
- XML Schema atbalsta datu tipus
  - Vieglāk aprakstīt atļauto dokumentu saturu
  - Vieglāk pārbaudīt datu pareizumu
  - Vieglāk apstrādāt datus no datu bāzēs
  - Vieglāk aprakstīt datu ierobežojumus un šablonus
  - Vieglāk konvertēt datus
- XML Schema tiek rakstīta XML valodā
  - Nav nepieciešamības mācīties jauno valodu
  - Var izmantot XML redaktoru un parsētāju
  - Var transformēt dokumenta shēmas aprakstu citā formātā

# 9. XML Schema standarts

- XML Schema standarts ir sadalīts trīs daļās:
  - XML Schema pamata koncepcijas
    - [www.w3.org/TR/xmlschema-0/](http://www.w3.org/TR/xmlschema-0/)
  - Struktūras, izmantotas XML Schemas
    - [www.w3.org/TR/xmlschema-1/](http://www.w3.org/TR/xmlschema-1/)
  - XML Schema data tipu apraksts
    - [www.w3.org/TR/xmlschema-2/](http://www.w3.org/TR/xmlschema-2/)

# 10. XML Schema documents (1)

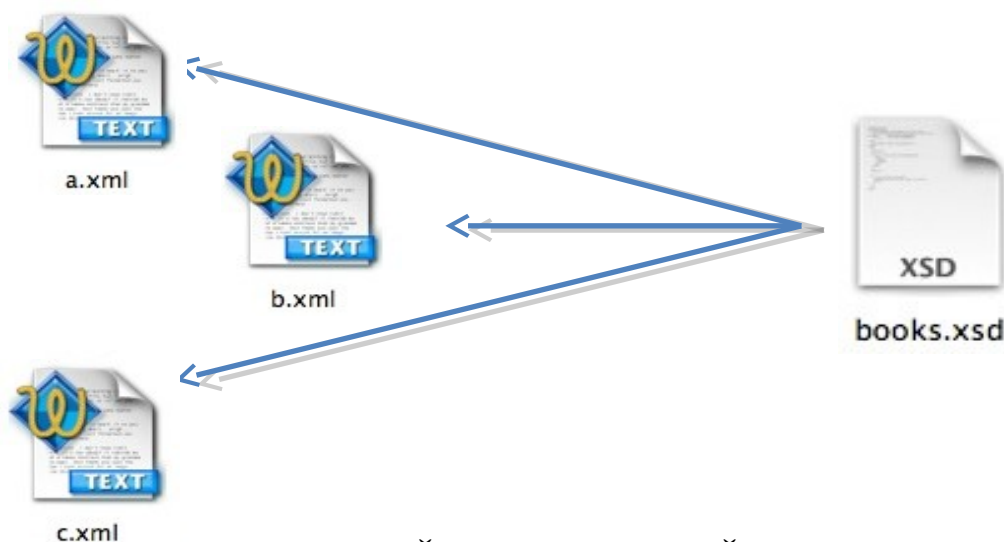
- XML Schema tiek saglabāta atsevišķajā XML dokumentā
- XML dokuments kurš ir uzbūvēts atbilstoši noteiktai XML Schema shēmas vārdnīcai tiek saukts par XML Schema instances dokumentu
- XML Schema faili parasti izmanto **.xsd** paplašinājumu
- “schema” sāknes elements



# 10. XML Schema documents (2)

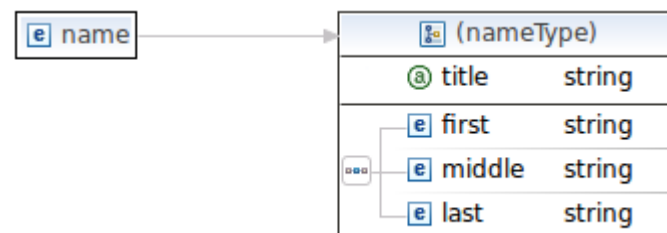
- XML dokumenti satur norādi uz XML Schema dokumentu, kas nosaka to vārdnīcu

```
<note xmlns="http://www.w3schools.com"  
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
      xsi:schemaLocation="http://www.w3schools.com note.xsd">
```



# 10. XML Schema documents (3)

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/NewXMLSchema"
  xmlns:tns="http://www.example.org/NewXMLSchema"
  elementFormDefault="qualified">
  <element name="name">
    <complexType>
      <sequence>
        <element name="first" type="string" />
        <element name="middle" type="string" />
        <element name="last" type="string" />
      </sequence>
      <attribute name="title" type="string" />
    </complexType>
  </element>
</schema>
```



# 11. <schema> elements (1)

- <schema> elements ir XML Schema dokumenta sāknes elements

**<?xml version="1.0"?>**

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

**...**

**</xs:schema>**



# 11. <schema> elements (2)

- <schema> elementā definējami papildus atribūti

```
<?xml version="1.0"?>
```

```
<xs:schema
```

```
xmlns:xs= "http://www.w3.org/2001/XMLSchema"
```

```
targetNamespace= "http://www.w3schools.com"
```

```
xmlns= "http://www.w3schools.com"
```

```
elementFormDefault= "qualified">
```

```
...
```

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

# 12. XML Schema vārdu telpa (1)

- Var izmantot jebkuru no variantiem

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

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

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

## 12. XML Schema vārdu telpa (2)

- Nosaka, ka elementi un datu tipi izmantoti XML shēmas definīcijā nāk no XMLSchema vārdnīcas  
(<http://www.w3.org/2001/XMLSchema>)
- Nosaka, ka elementi un datu tipi no “<http://www.w3.org/2001/XMLSchema>” jāraksta ar “xs:” vai “xsd:” prefiksu

# 13. Mērķa vārdu telpa (targetNamespace)

- Pirmatnēja XML Schema nozīme ir XML vārdnīcu deklarēšana vārdu telpai, specificētai “targetNamespace” atribūtā
- Deklarējot mērķa vārdu telpu ir svarīgi iekļaut atbilstošas deklarācijas

**<xs:schema**

**xmlns:xs="http://www.w3.org/2001/XMLSchema"**

**targetNamespace="http://www.example.org/NewXMLSchema"**

**xmlns:tns="http://www.example.org/NewXMLSchema">**



# 14. XML Schema datu tipi

- Atļauj vairāk specifikas par DTD
- Var norādīt datumus, numurus, diapazonus, unc



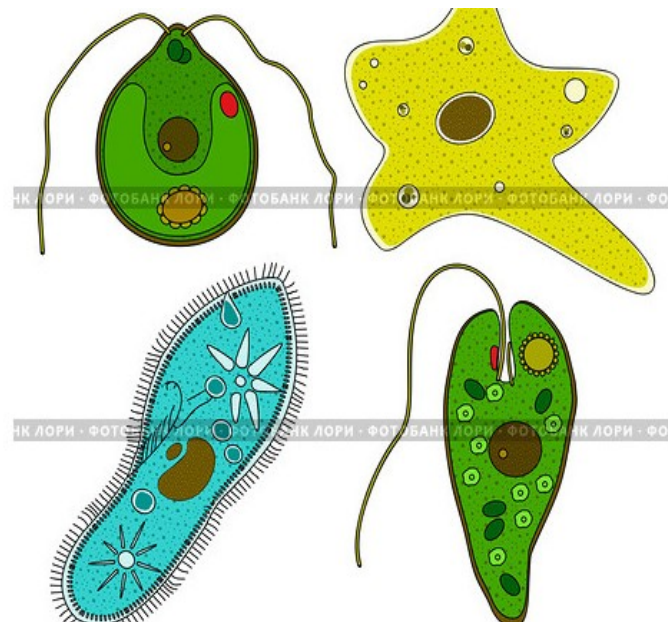
Var iedalīt divās kategorijās:

**Vienkāršie** - nodarbojas ar pamatvērtībām

**Sarežģītie** - apraksta sarežģītākas konstrukcijas

# 15. Vienkāršie datu tipi (1)

- Apraksta tekstu, ciparus, datumus
- Bieži sauktas par “primitīvajiem”
- Vienkāršo datu tipu elementiem nevar būt atribūtu
- Iebūvētie datu tipi ir XML Schema vārdu telpā



# 15. Vienkāršie datu tipi (2)

String

Boolean

Numbers

Integer

Decimal

Float

Double

Custom (simpleType)

Date/time

Time

Timestamp

Duration

Date

Month

Year

...

# 16. Vienkāršā <element> deklarēšana (1)

**<xs:element name="NAME" type="TYPE"/>**

- NAME – elementa nosaukums, jāseko XML noteikumiem elementu nosaukumiem
- TYPE – elementa datu tips
- XML Schema definē vairākus iebūvētus datu tipus

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

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

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



## 16. Vienkāršā <element> deklarēšana (2)

```
<xsd:element name="firstname"  
              type="xsd:string" />
```

```
<xsd:element name="ableToSwim"  
              type="xsd:boolean" />
```

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

## 16. Vienkāršā <element> deklarēšana (3)

- Kardinalitāte (angl. cardinality) nosaka cik reizes specifisks elements var parādīties satura modelī

**<element name="first" type="string"  
minOccurs="x" maxOccurs="y"/>**

- x : minimālais elementa parādīšanas skaits
- y : maksimālais elementa parādīšanas skaits

## 16. Vienkāršā <element> deklarēšana (4)

- Vienkāršiem elementiem XML shēmā var definēt vērtības pēc noklusēšanas vai nemainīgas vērtības
  - Vērtība pēc noklusēšanas tiek automātiski piešķirta elementam, kad netiek specificēta cita vērtība

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

- Nemainīgas vērtības tiek automātiski piešķirtas elementam. Elementa vērtību nevar izmainīt uz citu.

**<xs:elementname="color" type="xs:string" fixed="red"/>>**

# 17. Vienkāršais vs. sarežģītais

complex  
simple

- Vienkāršais
  - `<grade>7</grade>`
  - nav citu iekļauto elementu
- Sarežģītais
  - `<students><student>Jack</student></students>`
  - ir iekļautie elementi

# 18. Sarežģīta elementa deklarēšana (1)

- Sarežģīts XML elements (angl. complex element) satur citus elementus un/vai atribūtus
- Ir četri sarežģīto elementu tipi:
  - Tukšie elementi
  - Elementi, kuri ietver tikai elementus
  - Elementi tikai ar tekstu
  - Elementi, kuri ietver gan tekstu gan citus elementus
- NB! Katrs no tiem elementu tipiem var ietvert arī atribūtus



## 18. Sarežģīta element deklarēšana (2)

- Tukšie elementi
  - Sarežģīts XML elements, "product", kurš ir tukšs  
**<product pid="1345"/>**
- Sarežģīts XML elements "employee", kurš satur tikai citus elementus

**<employee>**

**<firstname>John</firstname>**

**<lastname>Smith</lastname>**

**</employee>**

## 18. Sarežģīta element deklarēšana (3)

- Sarežģīts XML elements "description", kurš satur gan tekstu gan citus elementus

```
<description> It happened on  
    <date lang="norwegian">03.03.99</date>  
</description>
```

# 19. <complexType> deklarēšana (1)

- "employee" elementu tipa deklarēšana

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/employee"
  xmlns:tns="http://www.example.org/employee"
  elementFormDefault="qualified">

  <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>

</xs:schema>
```



## 19. <complexType> deklarēšana (2)

- "employee" elementu tipa deklarēšana
  - Tikai "employee" elements var izmantot deklarēto sarežģītu tipu
  - Elementi "firstname" un "lastname" apvienoti zem viena <sequence> indikatora, kas nozīmē, ka dokumentā tiem jāparādās tādā pati kārtībā, kā deklarēti shēmā



## 20. <complexType> deklarēšana (3)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/employee"
  xmlns:tns="http://www.example.org/employee"
  elementFormDefault="qualified">

  <xs:element name="employee" type="tns:EmployeeType" />

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

  <xs:element name="department">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="employee"
          type="tns:EmployeeType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## 21. <attribute> deklarēšana

- XML atribūta definēšanas sintakse:

**<xs:attributename="ANAME"  
type="ATYPE"/>**

- ANAME – atribūta nosaukums
- ATYPE – atribūta datu tips
- Vienkāršajiem alementiem nevar būt atribūti

# 22. Kopsavilkums

- XML Schema ir W3C XML tehnoloģijas XML dokumentu shēmu definēšanai
- Apraksta XML dokumenta struktūru
- Apraksta sintakses un struktūras noteikumu kopu



# Papildus XSD piemēri

- Iebūvēto bāzes tipu pārveidošana
  - Vērtību ierobežojumi
  - Diapazoni
  - Saraksti

```
<xsd:simpleType="Degrees">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="AA" />  
    <xsd:enumeration value="AS" />  
  </xsd:restriction>  
</xsd:simpleType>
```

# Papildus XSD piemēri

- Saraksts (angl. enumeration)

```
<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>
```

# Papildus XSD piemēri

- Šablons ar REGEX

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

# Papildus XSD piemēri

## *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}" />
```



# Papildus XSD piemēri

## Elementu kārtībā

- **Sequence:** elements parādās tajā pašā kārtībā, kā ir definēts XML Schema
- **All:** elementi parādās jebkura kārdība

```
<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>
```

# Tukšs elements atribūtu

- 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>`

# Praktiskais uzdevums

