



samen sterk voor werk

XML fundamentals takenbundel

Deze cursus is eigendom van de VDAB©

Inhoudsopgave

1	TAKEN	3
1.1	Declaration en commentaar	3
1.2	Daltons	3
1.3	CD verzameling.....	3
1.4	CD verzameling 2.....	3
1.5	Quatre quarts	3
1.5.1	Ingrediënten.....	3
1.5.2	Bereiding	3
1.6	Boomstructuur	3
1.7	XPath	3
1.8	Namespaces	4
1.9	Regular expression	4
1.10	Schema zonder namespace.....	4
1.11	Schema met namespace	4
1.12	Attributen	4
1.13	Ingebouwde simple types	5
1.14	Zelf gedefinieerde simple types	5
1.15	Complex type.....	5
1.16	Bieren	5
2	VOORBEELDOPLOSSINGEN	6
2.1	Declaration en commentaar	6
2.2	Daltons	6
2.3	CD verzameling.....	6
2.4	CD verzameling 2.....	6
2.5	Quatre quarts	6
2.6	Boomstructuur	7

2.7	XPath	7
2.8	Namespaces	8
2.8.1	ComputerTalen.xml	8
2.8.2	GesprokenTalen.xml	8
2.8.3	Talen.xml	8
2.9	Regular expression	8
2.10	Schema zonder namespace	8
2.10.1	Artikel.xsd	8
2.10.2	Koffiezetmachine.xml	8
2.11	Schema met namespace	8
2.11.1	Artikel.xsd	8
2.11.2	Koffiezetmachine.xml	9
2.12	Attributen	9
2.12.1	Artikel.xsd	9
2.12.2	Koffiezetmachine.xml	9
2.13	Ingebouwde simple types	9
2.13.1	Artikel.xsd: gewijzigde regels	9
2.13.2	Koffiezetmachine.xml	9
2.14	Zelf gedefinieerde simple types	9
2.14.1	Artikel.xsd	9
2.14.2	Koffiezetmachine.xml	10
2.15	Complex type	10
2.15.1	Artikel.xsd: gewijzigd complexType	10
2.15.2	Koffiezetmachine.xml	10
2.16	Bieren	10
3	COLOFON	13

1 TAKEN

1.1 Declaration en commentaar

Corrigeer DeclarationEnCommentaar.xml tot het een well formed document is.

1.2 Daltons

Corrigeer Daltons.xml tot het een well formed document is.

1.3 CD verzameling

Vul CDVerzameling.xml aan met data over een CD verzameling met bijvoorbeeld drie CD's. Je houdt per CD de titel, de artiest en het jaar bij.

1.4 CD verzameling 2

Gebruik in CDVerzameling.xml zoveel mogelijk attributen.

1.5 Quatre quarts

Vul QuatreQuarts.xml aan met de ingrediënten en bereiding van een quatre quarts taart, gebaseerd op onderstaande tekst.

Je beslist zelf of je data opneemt in attributen of child elementen.

1.5.1 Ingrediënten

- 200 gram suiker
- 200 gram boter
- 200 gram bloem
- 4 eieren

1.5.2 Bereiding

1. Meng eerst de suiker met de boter. Voeg vervolgens het zout en twee eieren toe en daarna de helft van de bloem. Voeg het laatste ei of de laatste eieren en de resterende bloem toe. U moet een homogeen en luchtig deeg krijgen.
2. Meng het deeg goed met een elektrische klopper en blijf lang genoeg mengen telkens wanneer u een ingrediënt toevoegt.
3. Giet het deeg in een met boter ingevette springvorm waarvan de bodem is bedekt met bakpapier. Laat 30 minuten bakken in een op 150 °C voorverwarmde oven (T° 5). Neem de cake na 5 minuten uit de vorm en laat afkoelen op een rooster. (ongeveer 45 minuten baktijd voor een cakevorm).
4. U kunt uw quatre-quarts cake op verschillende manieren een persoonlijke noot geven: met sinaasappel- of citroenschil, stukjes chocolade of pure cacao, stukjes appel of peer (met kaneel), stukjes abrikoos of perzik, gekonfijt fruit, rozijnen, enzovoort.

1.6 Boomstructuur

Teken de boomstructuur van je oplossing in QuatreQuarts.xml

1.7 XPath

Welke XPath expressies heb je nodig om in Popgroep.xml volgende vragen te beantwoorden:

1. Wat zijn de voornamen van alle leden?
2. Wat zijn de voornamen van de vrouwen?
3. Wat is de laatst vermelde taak van Benny Andersson?
4. Wat zijn de voornamen van de leden geboren in 1945?
5. Wat zijn de voornamen van de leden die zingen?
6. Wat zijn de voornamen van de leden met meer dan één taak?

1.8 Namespaces

Associeer de elementen in ComputerTalen.xml met de namespace <http://www.computerlanguages.org/list>

Associeer de elementen in GesprokenTalen.xml met de namespace <http://www.spokenlanguages.org/list>

Associeer de juiste elementen in Talen.xml met de namespace <http://www.computerlanguages.org/list> of met de namespace <http://www.spokenlanguages.org/list>

1.9 Regular expression

Je maakt een regular expression voor een e-mail adres

- Het begint met één of meerdere tekens (om het even welke)
- Daarna komt het teken @
- Daarna volgen één of meerdere tekens (om het even welke)
- Daarna komt een punt
- Daarna volgen één of meerdere tekens (om het even welke)

Je maakt een regular expression voor een oud Belgisch bankrekeningnummer

- Het begint met 3 cijfers
- Daarna komt het teken –
- Daarna komen 7 cijfers
- Daarna komt het teken –
- Daarna komen 2 cijfers

Je maakt een regular expression voor een nummerplaat

- Het begint met 1 cijfer
- Daarna komt het teken –
- Daarna komen 3 letters tussen A en Z
- Daarna komt het teken –
- Daarna komen 3 cijfers

1.10 Schema zonder namespace

1. Je maakt een kopie van Persoon.xsd als Artikel.xsd
2. Je wijzigt dit schema: het definieert dat de naam van root element artikel moet zijn.
3. Je maakt een kopie van Jean.xml als Koffiezetmachine.xml
4. Je verwijst hierin naar Artikel.xsd
5. Je wijzigt het root element naar artikel
6. Je controleert of de XML well formed en valid is.

1.11 Schema met namespace

1. Je associeert het schema in Artikel.xsd met de namespace <http://www.vdab.be/artikels>
2. Je associeert de XML in Koffiezetmachine.xml met dezelfde namespace en je doet de nodige aanpassingen zodat de XML well formed en valid is.

1.12 Attributen

1. Je definieert in Artikel.xsd dat een artikel volgende verplichte attributen heeft
 - a. productieDatum
 - b. prijs
2. Je definieert dat een artikel volgende optionele attributen heeft
 - a. lengte
 - b. breedte
 - c. hoogte
3. Je wijzigt Koffiezetmachine.xml zodat de XML well formed en valid is.

1.13 Ingebouwde simple types

1. Je geeft in Artikel.xsd elk attribuut het best bijbehorende simple type
2. Je wijzigt Koffiezetmachine.xml zodat de XML well formed en valid is

1.14 Zelf gedefinieerde simple types

1. Het attribuut productieDatum moet minstens 1 januari 2000 zijn
2. Het attribuut prijs moet minstens 0 zijn en bevat maximaal 2 decimalen
3. De attributen lengte, breedte en hoogte moeten groter dan 0 zijn

1.15 Complex type

1. Je wijzigt in Artikel.xsd attributen naar child elementen, waarbij de volgorde van die child elementen niet belangrijk is
2. Je wijzigt in Koffiezetmachine.xml attributen naar child elementen

1.16 Bieren

Je maakt een schema dat past bij Bieren.xml

- Het element bieren kan 0 tot oneindig aantal bier elementen bevatten
- In bier is biernr verplicht, bevat een positief geheel getal en is uniek per bier
- In bier is naam verplicht
- In bier is brouwnr verplicht, bevat een positief geheel getal en verwijst naar een brouwnr van een brouwerij element (verder in het bestand)
- In bier is soortnr verplicht, bevat een positief geheel getal en verwijst naar een soortnr van een soort element (verder in het bestand)
- In bier is alcohol optioneel. Als het voorkomt bevat het een niet-negatief getal.
- In brouwerij is brouwnr verplicht, bevat een positief geheel getal en is uniek per brouwerij
- In brouwerij is naam verplicht
- In brouwerij is adres verplicht
- In brouwerij is postcode verplicht en bevat een geheel getal tussen 1000 en 9999
- In brouwerij is omzet optioneel. Als het voorkomt bevat het een niet-negatief getal
- In soort is soortnr verplicht, bevat een positief geheel getal en is uniek per soort
- In soort is naam verplicht

2 VOORBEELDOPLOSSINGEN

2.1 Declaration en commentaar

```
<?xml version="1.0" encoding="utf-8"?>
<eersteTaak/>
<!--
Maak hiervan
een well formed document
-->
```

2.2 Daltons

```
<?xml version="1.0" encoding="utf-8"?>
<daltons>
  <name>Joe</name>
  <name>Jack</name>
  <name>William</name>
  <name>Averell</name>
</daltons>
```

2.3 CD verzameling

```
<?xml version="1.0" encoding="utf-8"?>
<cdverzameling>
  <cd>
    <titel>Thin walls</titel>
    <artiest>Balthazar</artiest>
    <jaar>2015</jaar>
  </cd>
  <cd>
    <titel>This Marauder's Midnight</titel>
    <artiest>Gabriel Rios</artiest>
    <jaar>2014</jaar>
  </cd>
  <cd>
    <titel>Kill all kings</titel>
    <artiest>Channel Zero</artiest>
    <jaar>2014</jaar>
  </cd>
</cdverzameling>
```

2.4 CD verzameling 2

```
<?xml version="1.0" encoding="utf-8"?>
<cdverzameling>
  <cd titel="Thin walls" artiest="Balthazar" jaar="2015"/>
  <cd titel="This Marauder's Midnight" artiest="Gabriel Rios" jaar="2014"/>
  <cd titel="Kill all kings" artiest="Channel Zero" jaar="2014"/>
</cdverzameling>
```

2.5 Quatre quarts

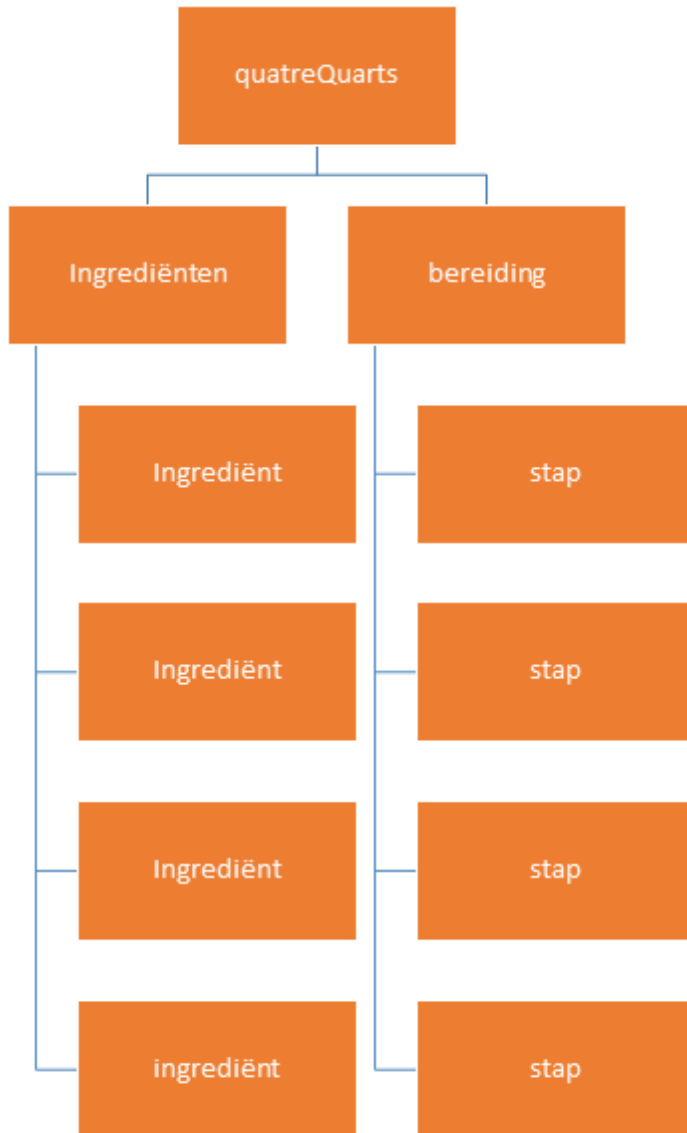
```
<?xml version="1.0" encoding="utf-8"?>
<quatreQuarts>
  <ingrediënten>
    <ingrediënt naam="suiker" hoeveelheid="200" eenheid="gram"/>
    <ingrediënt naam="boter" hoeveelheid="200" eenheid="gram"/>
    <ingrediënt naam="bloem" hoeveelheid="200" eenheid="gram"/>
    <ingrediënt naam="ei" hoeveelheid="4" eenheid="stuk"/>
  </ingrediënten>
</quatreQuarts>
```

```

<bereiding>
  <stap>Meng eerst de suiker met de boter. ...</stap>
  <stap>Meng het deeg goed met een elektrische klopper ...</stap>
  <stap>Giet het deeg in een met boter ingevette springvorm ...</stap>
  <stap>U kunt uw quatre-quarts cake op verschillende manieren ...</stap>
</bereiding>
</quatreQuarts>

```

2.6 Boomstructuur



2.7 XPath

1. `//lid/@voornaam`
2. `//lid[@geslacht="V"]/@voornaam`
3. `//lid[@voornaam="Benny" and @familienaam="Andersson"]/taken/taak[last()]`
4. `//lid[geboortejaar="1945"]/@voornaam`
5. `//lid[taken/taak="zang"]/@voornaam`
6. `//lid[count(taken/taak)>1]/@voornaam`

2.8 Namespaces

2.8.1 ComputerTalen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<talen xmlns="http://www.computerlanguages.org/List">
  <taal>Java</taal>
  <taal>C#</taal>
  <taal>PHP</taal>
</talen>
```

2.8.2 GesprokenTalen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<talen xmlns="http://www.spokenlanguages.org/List">
  <taal>Nederlands</taal>
  <taal>Frans</taal>
  <taal>Duits</taal>
</talen>
```

2.8.3 Talen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<talen xmlns:pc="http://www.computerlanguages.org/List"
      xmlns:hu="http://www.spokenlanguages.org/List">
  <pc:taal>Java</pc:taal>
  <pc:taal>C#</pc:taal>
  <pc:taal>PHP</pc:taal>
  <hu:taal>Nederlands</hu:taal>
  <hu:taal>Frans</hu:taal>
  <hu:taal>Duits</hu:taal>
</talen>
```

2.9 Regular expression

e-mail adres:

`^.+@.+\.+.$`

oud Belgisch bankrekeningnummer:

`^\d{3}-\d{7}-\d{2}$`

nummerplaat:

`^\d-[A-Z]{3}-\d{3}$`

2.10 Schema zonder namespace

2.10.1 Artikel.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="artikel"/>
</schema>
```

2.10.2 Koffiezetmachine.xml

```
<?xml version="1.0" encoding="utf-8"?>
<artikel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="Artikel.xsd"/>
```

2.11 Schema met namespace

2.11.1 Artikel.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://www.vdab.be/artikels">
  <element name="artikel"/>
</schema>
```

2.11.2 Koffiezetmachine.xml

```
<?xml version="1.0" encoding="utf-8"?>
<artikel xmlns="http://www.vdab.be/artikels"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.vdab.be/artikels Artikel.xsd"/>
```

2.12 Attributen

2.12.1 Artikel.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:vdab="http://www.vdab.be/artikels"
  targetNamespace="http://www.vdab.be/artikels">
  <complexType name="Artikel">
    <attribute name="productieDatum" use="required"/>
    <attribute name="prijs" use="required"/>
    <attribute name="Lengte"/>
    <attribute name="breedte"/>
    <attribute name="hoogte"/>
  </complexType>
  <element name="artikel" type="vdab:Artikel"/>
</schema>
```

2.12.2 Koffiezetmachine.xml

```
<?xml version="1.0" encoding="utf-8"?>
<artikel xmlns="http://www.vdab.be/artikels"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.vdab.be/artikels Artikel.xsd"
  productieDatum="2013-02-14" prijs="40.33"/>
```

2.13 Ingebouwde simple types

2.13.1 Artikel.xsd: gewijzigde regels

```
<attribute name="productieDatum" use="required" type="date"/>
<attribute name="prijs" use="required" type="decimal"/>
<attribute name="Lengte" type="decimal"/>
<attribute name="breedte" type="decimal"/>
<attribute name="hoogte" type="decimal"/>
```

2.13.2 Koffiezetmachine.xml

De vorige voorbeeldoplossing blijft well formed en valid

2.14 Zelf gedefinieerde simple types

2.14.1 Artikel.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:vdab="http://www.vdab.be/artikels"
  targetNamespace="http://www.vdab.be/artikels">
  <simpleType name="ProductieDatum">
    <restriction base="date">
      <minInclusive value="2000-01-01"/>
    </restriction>
  </simpleType>
  <simpleType name="Prijs">
    <restriction base="decimal">
      <minInclusive value="0"/>
      <fractionDigits value="2"/>
    </restriction>
  </simpleType>
  <simpleType name="Afmeting">
    <restriction base="decimal">
      <minExclusive value="0"/>
    </restriction>
  </simpleType>
</schema>
```

```

    </restriction>
  </simpleType>
  <complexType name="Artikel">
    <attribute name="productieDatum" use="required" type="vdab:ProductieDatum"/>
    <attribute name="prijs" use="required" type="vdab:Prijs"/>
    <attribute name="lengte" type="vdab:Afmeting"/>
    <attribute name="breedte" type="vdab:Afmeting"/>
    <attribute name="hoogte" type="vdab:Afmeting"/>
  </complexType>
  <element name="artikel" type="vdab:Artikel"/>
</schema>

```

2.14.2 Koffiezetmachine.xml

De vorige voorbeeldoplossing blijft well formed en valid

2.15 Complex type

2.15.1 Artikel.xsd: elementFormDefault="qualified" toegevoegd en complexType gewijzigd

```

<complexType name="Artikel">
  <all>
    <element name="productieDatum" type="vdab:ProductieDatum"/>
    <element name="prijs" type="vdab:Prijs"/>
    <element name="lengte" type="vdab:Afmeting" minOccurs="0"/>
    <element name="breedte" type="vdab:Afmeting" minOccurs="0"/>
    <element name="hoogte" type="vdab:Afmeting" minOccurs="0"/>
  </all>
</complexType>

```

2.15.2 Koffiezetmachine.xml

```

<?xml version="1.0" encoding="utf-8"?>
<artikel xmlns="http://www.vdab.be/artikels"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.vdab.be/artikels Artikel.xsd">
  <productieDatum>2013-02-14</productieDatum>
  <prijs>40.33</prijs>
</artikel>

```

2.16 Bieren

```

<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:vdab="http://www.vdab.be/bieren"
  targetNamespace="http://www.vdab.be/bieren" elementFormDefault="qualified">
  <!-- attributen
  =====
  biernr : positiveInteger
  brouwnr : positiveInteger
  soortnr : positiveInteger -->
  <!-- elementen met enkel een text node
  =====
  naam : string
  brouwnr : positiveInteger
  soortnr : positiveInteger
  alcohol : Alcohol (zelf gedefinieerd)
  naam : string
  adres : string
  postcode : Postcode (zelf gedefinieerd)
  gemeente : string
  omzet : nonNegativeInteger
  Soort : string -->

```

```

<simpleType name="Postcode">
  <restriction base="positiveInteger">
    <minInclusive value="1000"/>
    <maxInclusive value="9999"/>
  </restriction>
</simpleType>
<simpleType name="Alcohol">
  <restriction base="decimal">
    <minInclusive value="0"/>
  </restriction>
</simpleType>
<complexType name="Bier">
  <sequence>
    <element name="naam" type="string"/>
    <element name="brouwernr" type="positiveInteger"/>
    <element name="soortnr" type="positiveInteger"/>
    <element name="alcohol" type="vdab:Alcohol" minOccurs="0"/>
  </sequence>
  <attribute name="biernr" type="positiveInteger" use="required"/>
</complexType>
<complexType name="Brouwer">
  <sequence>
    <element name="naam" type="string"/>
    <element name="adres" type="string"/>
    <element name="postcode" type="vdab:Postcode"/>
    <element name="gemeente" type="string"/>
    <element name="omzet" type="nonNegativeInteger" minOccurs="0"/>
  </sequence>
  <attribute name="brouwernr" type="positiveInteger" use="required"/>
</complexType>
<complexType name="Soort">
  <sequence>
    <element name="naam" type="string"/>
  </sequence>
  <attribute name="soortnr" type="positiveInteger" use="required"/>
</complexType>
<complexType name="Bieren">
  <sequence>
    <element name="bier" type="vdab:Bier" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="Brouwers">
  <sequence>
    <element name="brouwerij" type="vdab:Brouwer" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="Soorten">
  <sequence>
    <element name="soort" type="vdab:Soort" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

```

```
<complexType name="BierenDB">
  <sequence>
    <element name="bieren" type="vdab:Bieren"/>
    <element name="brouwers" type="vdab:Brouwers"/>
    <element name="soorten" type="vdab:Soorten"/>
  </sequence>
</complexType>
<element name="bierendb" type="vdab:BierenDB">
  <key name="bierKey">
    <selector xpath="vdab:bieren/vdab:bier"/>
    <field xpath="@biernr"/>
  </key>
  <key name="brouwerKey">
    <selector xpath="vdab:brouwers/vdab:brouwerij"/>
    <field xpath="@brouwernr"/>
  </key>
  <key name="soortKey">
    <selector xpath="vdab:soorten/vdab:soort"/>
    <field xpath="@soortnr"/>
  </key>
  <keyref name="bier_brouwer" refer="vdab:brouwerKey">
    <selector xpath="vdab:bieren/vdab:bier"/>
    <field xpath="vdab:brouwernr"/>
  </keyref>
  <keyref name="bier_soort" refer="vdab:soortKey">
    <selector xpath="vdab:bieren/vdab:bier"/>
    <field xpath="vdab:soortnr"/>
  </keyref>
</element>
</schema>
```

3 COLOFON

Domeinexpertisemanager:	Jean Smits
Moduleverantwoordelijke:	Hans Desmet
Medewerkers:	Hans Desmet
Versie:	16/9/2015
Nummer dotatielijst:	