

XSL Transformations (XSLT)

Version 1.0, Recommendation, 16.11.1999

The diagram illustrates the XSLT process. On the left, a box labeled 'henkilosto.xml' contains XML code: `<henkilosto>`, `...`, and `</henkilosto>`. Three arrows point from this box to the right. The top arrow points to a box labeled 'tikohenk.xml' containing `<henkilosto>`, `...`, and `</henkilosto>`. The middle arrow points to a laptop icon. The bottom arrow points to a mobile phone icon. Below the diagram, there is explanatory text.

XSLT:ä (XSL Transformation) käytetään muuntamaan XML-dokumentti toiseksi XML-, XHTML-, HTML- tai tekstidokumentiksi.

XSL:illa eli XSL-FO:lla (XSL Formating Object) tuotetaan dokumentteja muuhun muotoon esim. pdf:ksi tai rtf:ksi.

1

XSLT

The flowchart shows the XSLT process. On the left, two boxes represent input: 'henkilosto.xslt' (described as 'Kuinka XML dokumentti muunnetaan toiseen muotoon') and 'henkilosto.xml' (described as 'Muunnettava XML-dokumentti'). Arrows from both boxes point to a central box labeled 'XSLT prosessori' (described as 'Ohjelma, joka muuntaa XML-dokumentin toiseksi XML-dokumentiksi XSLT-dokumentin ohjeiden mukaan'). An arrow from the processor box points to a final box on the right labeled 'henkilosto.html' (described as 'Muunnoksen tulos').

2

XML-dokumentti

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<henkilosto>
  <henkilo tunnus="mars">
    <nimi>
      <sukunimi entinen="Lehtonen">Marttila</sukunimi>
      <etunimi>Sirpa</etunimi>
      <etunimi>Anneli</etunimi>
    </nimi>
    <email>sirpa.marttila@helia.fi</email>
    <puhelin type="työ">09-14890270</puhelin>
  </henkilo>
  <henkilo tunnus="honte">
    <nimi>
      <sukunimi>Honkasaari</sukunimi>
      <etunimi>Terttu</etunimi>
    </nimi>
    <email>terttu.honkasaari@helia.fi</email>
    <puhelin>09-148901</puhelin>
    <puhelin>040-7654321</puhelin>
  </henkilo>
</henkilosto>
```

3

XSLT-dokumentti

```
<?XML version="1.0" encoding="ISO-9959-1"?>
<xsl:stylesheet version="1.0" xmlns="http://www.w3.org/1999/xhtml"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
encoding="ISO-8859-1" method="xml" version="1.0"/>
<xsl:template match="/">
<html>
<head>
<title> Helian henkilöstö </title>
</head>
<body>
<h2> Henkilöstö </h2>
<xsl:apply-templates select="henkilosto/henkilo"/>
</body>
</html>
</xsl:template>

<xsl:template match="henkilo">
Tunnus: <xsl:value-of select="@tunnus"/><br />
</xsl:template>
```

4

- ▼ XSLT-dokumentti noudattaa XML-muotoa
- ▼ XSLT-dokumentti on hyvin muodostettu (well-formed) XML-dokumentti. XSLT-dokumentissa
 - on yksi juurielementti: `<xsl:stylesheet ...>`.
 - elementit ovat sisäkkäisiä.
 - jokaisella elementillä on alku- ja lopputunniste (huom: `
` ja `<hr />`).
 - suurilla ja pienillä kirjaimilla on merkitystä. `<head>` on eri asia kuin `<HEAD>`.
 - attribuuttien arvot ovat lainaus- tai heittomerkeissä.

5

Juurielementti

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<xsl:stylesheet version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
xmlns="http://www.w3.org/1999/xhtml">  
</xsl:stylesheet>
```

- ▼ *xsl* on xsl-dokumenttien nimiavaruus. XSLT- prosessori tulkitsee XSLT-dokumentista vain xsl-alkuiset elementit.

6

Tulosdokumentin tyypin määrittely

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml" >
<xsl:output method="xml" version="1.0" encoding="ISO-8859-1"
doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" />
</xsl:stylesheet>
```

- ▼ *xsl:output* lause kertoo, mihin muotoon (xml, html, text) muunnokset tehdään sekä tulosdokumentin version, koodauksen sekä mahdollisen rakennemäärittelyn (XML:n yhteydessä).

7

XML

```
<xsl:output method="xml" version="1.0" encoding="ISO-8859-1"
omit-xml-declaration="no"
doctype-system="henkilosto.dtd"/>
```

XSLT-dokumentti

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE juuri SYSTEM "henkilosto.dtd">
```

tulosdokumentti

XHTML

```
<xsl:output method="xml" version="1.0" encoding="ISO-8859-1"
doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" />
```

XSLT-dokumentti

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

tulosdokumentti

8

Malli (template)

- ▼ ideana on kirjoittaa elementtien käsittelyyn mallisääntöjä, jotka kuvaavat, kuinka XML-dokumenttia käsitellään.

```
<xsl:template match="malli">  
  <!-- XML-dokumentin käsittelyä -->  
</xsl:template>
```

- ▼ mallille voidaan kertoa, mitä muita mallisääntöjä käytetään.

```
<xsl:template match="malli_1">  
  <xsl:apply-templates select="malli_2" />  
</xsl:template>
```

9

```
<xsl:template match="/">
```

```
  <xsl:apply-templates select="henkilosto/henkilo" />  
</xsl:template>
```

```
<xsl:template match="henkilo">  
  Tunnus: <br />  
</xsl:template>
```

XML-dokumentista
haetaan kaikki henkilosto-
elementin henkilo-
lapsielementit

henkilo-elementistä
kirjoitetaan tulodoku-
menttiin Tunnus ja


```
<henkilosto>  
  <henkilo tunnus="mars">  
    <email>sirpa.marttila@helia.fi</email>  
  </henkilo>  
  <henkilo tunnus="honte">  
    <email>terttu.honkasaari@helia.fi</email>  
  </henkilo>  
</henkilosto>
```

Tunnus:
Tunnus:

10

Arvon hakeminen XML-dokumentista

- ▼ Malliin voidaan kirjoittaa tekstiä, joka menee sellaisenaan tulodokumenttiin.
- ▼ XML-dokumentista haetaan tulodokumenttiin tietoa value-of -elementillä, mihin laitetaan XPath-lauseke haettavalle tiedolle.

```
<xsl:template match="malli_2">  
  Nimi:  
  <xsl:value-of select="haettava_tieto" />  
</xsl:template>
```

11

```
<xsl:template match="/">  
  <xsl:apply-templates select="henkilosto/henkilo" />  
</xsl:template>  
  
<xsl:template match="henkilo">  
  Tunnus: <xsl:value-of select="@tunnus" /> <br />  
  Email: <xsl:value-of select="email" /> <br />  
</xsl:template>
```

Teksti tulostuu tulodokumenttiin sellaisenaan

XML-dokumentista haetaan tulodokumenttiin henkilö-elementin tunnus-attribuutti

```
<henkilosto>  
<henkilo tunnus="mars">  
  <email>sirpa.marttila@helia.fi</email>  
</henkilo>  
<henkilo tunnus="honte">  
  <email>terttu.honkasaari@helia.fi</email>  
</henkilo>  
</henkilosto>
```

Tunnus: marsi
Email: sirpa.marttila@helia.fi
Tunnus: honte
Email: terttu.honkasaari@helia.fi

12

Yhden vs. usean arvon hakeminen

```
<xsl:template match="/">
  <xsl:apply-templates select="henkilosto/henkilo" />
</xsl:template>

<xsl:template match="henkilo">
  Puhelimet: <xsl:value-of select="puhelin" />
</xsl:template>
```

XML-dokumentista haetaan tulosdokumenttiin henkilö-elementin ensimmäinen puhelin-lapsielementti (ei muita !)

```
<henkilosto>
  <henkilo tunnus="mars">
    <puhelin>09-14890270</puhelin>
  </henkilo>
  <henkilo tunnus="hont">
    <puhelin>09-148901</puhelin>
    <puhelin>040-7654321</puhelin>
  </henkilo>
</henkilosto>
```

Puhelimet: 09-14890270
Puhelimet: 09-148901

=> hakee vain yhden puhelimen kullakin henkilöltä

13

```
<xsl:template match="/">
  <xsl:apply-templates select="henkilosto/henkilo" />
</xsl:template>

<xsl:template match="henkilo">
  Puhelimet: <xsl:apply-templates select="puhelin" />
</xsl:template>
```

Puhelin-elementille ei ole kirjoitettu mallisääntöä, jolloin XSLT käyttää oletusmallisääntöä, mikä kirjoittaa haettavan elementin

```
<henkilosto>
  <henkilo tunnus="mars">
    <puhelin>09-14890270</puhelin>
  </henkilo>
  <henkilo tunnus="hont">
    <puhelin>09-148901</puhelin>
    <puhelin>040-7654321</puhelin>
  </henkilo>
</henkilosto>
```

Puhelimet: 09-14890270
Puhelimet: 09-148901040-7654321

=>kirjoittaa puhelimet kiinni toisiinsa

14

```
<xsl:template match="/">
  <xsl:apply-templates select="henkilosto/henkilo" />
</xsl:template>

<xsl:template match="henkilo">
  Puhelimet: <xsl:apply-templates select="puhelin" />
</xsl:template>

<xsl:template match="puhelin">
  <xsl:value-of select="." />&#160;
</xsl:template>
```

Käytetään itse tehtyä mallisääntöä, mikä kirjoittaa elementin tekstisisällön jälkeen tyhjälöyönnin

```
<henkilosto>
  <henkilo tunnus="mars">
    <puhelin>09-14890270</puhelin>
  </henkilo>
  <henkilo tunnus="honte">
    <puhelin>09-148901</puhelin>
    <puhelin>040-7654321</puhelin>
  </henkilo>
</henkilosto>
```

Puhelimet: 09-14890270
Puhelimet: 09-148901 040-7654321

15

Lajittelu

```
<xsl:apply-templates select="malli">
  <xsl:sort select="lajittelu"
    data-type="text" | "number"
    order="ascending" | "descending"
    case-order="upper-first" | "lower-first" />
</xsl:apply-templates>
```

16

```
<xsl:template match="/">
  <xsl:apply-templates select="henkilosto/henkilo" >
    <xsl:sort select="@tunnus" />
  </xsl:apply-templates>
</xsl:template>

<xsl:template match="henkilo">
  Tunnus: <xsl:value-of select="@tunnus" /> <br />
  Email: <xsl:value-of select="email" /> <br />
</xsl:template>
```

Lajitellaan henkilö-
elementit tunnus-
attribuuttien perusteella

```
<henkilosto>
  <henkilo tunnus="mars">
    <email>sirpa.marttila@helia.fi</email>
  </henkilo>
  <henkilo tunnus="honte">
    <email>terttu.honkasaari@helia.fi</email>
  </henkilo>
</henkilosto>
```

Tunnus: honte
Email: terttu.honkasaari@helia.fi
Tunnus: marsi
Email: sirpa.marttila@helia.fi

17

Parametrien käyttö

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:param name="param_nimi" select="oletusarvo" />
  ...
</xsl:stylesheet>
```

\$param_nimi

18

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:param name="sukun" />
<xsl:template match="/">
  <xsl:value-of select="$sukun" /> <br />
  <xsl:apply-templates
    select="henkilosto/henkilo/nimi[sukunimi=$sukun]/etunimi" />
</xsl:template>
</xsl:stylesheet>
```

XSLT:lle voidaan välittää käynnistysparametreja. Tässä välitetään haettavan henkilön sukunimi

```
<henkilosto>
<henkilo tunnus="mars">
  <nimi>
    <sukunimi>Marttila</sukunimi>
    <etunimi>Sirpa</etunimi>
  </nimi>
</henkilo>
</henkilosto>
```

Marttila
Sirpa

19

Valinta

```
<xsl:if test="ehto">...</xsl:if>
```

```
<xsl:choose>
  <xsl:when test="ehto1"> ...</xsl:when>
  <xsl:when test="ehto2"> ...</xsl:when>
  <xsl:otherwise> ...</xsl:otherwise >
</xsl:choose>
```

20

```

<xsl:template match="henkilo">
  Puhelimet: <xsl:apply-templates select="puhelin" />
</xsl:template>
<xsl:template match="puhelin">
  <xsl:value-of select="."/>
  <xsl:if test="position() &lt; last()">, <br /></xsl:if>
</xsl:template>

```

Jos puhelin-elementti ei ole viimeinen, kirjoitetaan tulosdokumenttiin pilkku. < tarkoittaa <. xsl:if rakenne ei sisällä else lausetta.

```

<henkilosto>
<henkilo tunnus="mars">
  <puhelin>09-14890270</puhelin>
</henkilo>
<henkilo tunnus="honte">
  <puhelin>09-148901</puhelin>
  <puhelin>040-7654321</puhelin>
</henkilo>
</henkilosto>

```

Puhelimet: 09-14890270
Puhelimet: 09-148901,
040-7654321

21

```

<xsl:template match="henkilo">
  <xsl:value-of select="tunnus" />&#160;
  <xsl:choose>
    <xsl:when test="kuva">Henkilöstä on kaunis kuva ...</xsl:when>
    <xsl:otherwise>Ei ole kuvaa ...</xsl:otherwise>
  </xsl:choose>
  <br />
</xsl:template>

```

Jos henkilo-elementillä on kuva-lapsielementti

Muussa tapauksessa

```

<henkilosto>
<henkilo tunnus="mars">
  <kuva url="sirpa" type="gif">
</henkilo>
<henkilo tunnus="honte">
</henkilo>
</henkilosto>

```

marsi Henkilöstä on kaunis kuva ...
honte Ei ole kuvaa ...

22

Toisesta dokumentista haku

- ▼ Tulodokumenttiin voidaan hakea tietoa useasta XML-dokumentista

- ▼ Yksi arvo

```
<xsl:template match="malli">  
  <xsl:value-of select="document('doku.xml')/hakupolku" />  
</xsl:template>
```

- ▼ Usea arvo

```
<xsl:template match="malli">  
  <xsl:apply-templates  
    select="document('doku.xml')/hakupolku" />  
</xsl:template>
```

23

```
<xsl:template match="/">  
  <xsl:apply-templates select="henkilosto/henkilo" />  
</xsl:template>  
  
<xsl:template match="henkilo">  
  <xsl:value-of select="nimi/sukunimi" />  
  <xsl:apply-templates  
    select="document('opetus.xml')/opetukset/henkilo[@tunnus='marsin']/opetus" />  
</xsl:template>
```

opetus.xml-dokumentista haetaan tulodokumenttiin tunnuksella marsi olevan henkilön opetus-elementtien sisällöt

```
<opetukset>  
  <henkilo tunnus="marsin">  
    <opetus>Atka3d</opetus>  
    <opetus>Atk28d</opetus>  
  </henkilo>  
  <henkilo tunnus="honte">  
    <opetus>AtkXYd</opetus>  
  </henkilo>  
</opetukset>
```

opetus.xml

```
Marttila  
Atk3d  
Atk28d  
Honkasaari => hakee kaikille marsin  
Atk3d opetuksen, koska select  
Atk28d lauseessa @tunnus='marsin'
```

24

Muuttujien käyttö

- ▼ muuttujien sisällöksi voidaan antaa ”kovakoodattu” arvo tai XML-dokumentista haettu arvo

- ▼ kovakoodattu arvo

```
<xsl:variable name="hae" select=" 'mars'" />
```

- ▼ arvon hakeminen XML-dokumentista

```
<xsl:variable name="hae" select=" @tunnus" />
```

- ▼ muuttujan arvoon viittaaminen

```
$hae
```

25

```
<xsl:template match="/">
  <xsl:apply-templates select="henkilosto/henkilo" />
</xsl:template>

<xsl:template match="henkilo">
  <xsl:value-of select="nimi/sukunimi" />
  <xsl:variable name="hae" select="@tunnus" />
  <xsl:apply-templates
    select="document('opetus.xml')/opetukset/henkilo[@tunnus=$hae]/opetus"/>
</xsl:template>
```

hae-muuttujaan laitetaan
vuorollaan kunkin
henkilön tunnus

```
<opetukset>
  <henkilo tunnus="mars">
    <opetus>Atka3d</opetus>
    <opetus>Atk28d</opetus>
  </henkilo>
  <henkilo tunnus="honte">
    <opetus>AtkXYd</opetus>
  </henkilo>
</opetukset>
```

opetus.xml

Marttila	
Atka3d	opetus.xml-dokumentista
Atk28d	haetaan tulosdokumenttiin
Honkasaari	kunkin henkilön
AtkXYd	tunnuksen perusteella
	hänen opetuksensa

26

Elementtien ja attribuuttien luonti

```
<xsl:templates match="malli">  
  <elementi attribuutti="{A-Sisältö}">E_Sisältö</elementi>  
</xsl:templates>
```

```
<elementi attribuutti="...">E_Sisältö</elementi>
```

Tai

```
<xsl:templates match="malli">  
  <xsl:element name="elementi">  
    <xsl:attribute name="attribuutti">A_Sisältö</xsl:attribute>  
    E_Sisältö  
  </xsl:element>  
</xsl:templates>
```

```
<elementi attribuutti="...">E_Sisältö</elementi>
```

27

```
<xsl:template match="henkilo">  
  <hlo tunnus="{@tunnus}"  
    etunimi="{nimi/etunimi}"  
    sukunimi="{nimi/sukunimi}">  
  </hlo>  
</xsl:template>
```

Luodaan tulosdokumenttiin hlo-elementti ja sille tunnus-, etunimi- ja sukunimi-attribuutit

```
<henkilosto>  
  <henkilo tunnus="mars">  
    <nimi>  
      <sukunimi>Marttila</sukunimi>  
      <etunimi>Sirpa</etunimi>  
    </nimi>  
  </henkilo>  
  <henkilo tunnus="honte">  
    <nimi>  
      <sukunimi>Honkasaari</sukunimi>  
      <etunimi>Terttu</etunimi>  
    </nimi>  
  </henkilo>  
</henkilosto>
```

```
<hlo sukunimi="Marttila"  
  etunimi="Sirpa"  
  tunnus="mars" />  
<hlo sukunimi="Honkasaari"  
  etunimi="Terttu"  
  tunnus="honte" />
```

28