




ELEC-C8203 Automaatiojärjestelmät 2 - XML

XPath kyselykieli

Pekka Aarnio

Sisältö

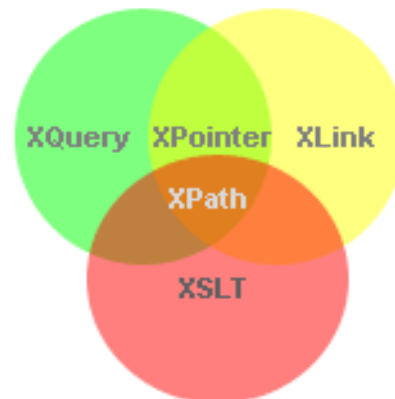
- 
- XPath lausekkeet
 - XPath joukko-oppi
 - XPath askeleet
 - XPath funktiot

XPATH

Online xpath tool: <http://www.xpathtester.com/xpath>

XPathin tehtävä

- XPath (XML Path Language) on kyselykieli, jolla XML-dokumentista voidaan poimia yksittäisiä elementtejä, attribuutteja tai jopa dokumentin osia
 - Näitä saatuja tuloksia joko käytetään hyväkseen suoraan tai lähtökohtana erilaisille toimenpiteille, esimerkiksi XSLT-muunnoksille (ks. XSLT luento)
- XPath on tärkeä koska monet XML-tekniikat hyödyntävät sitä



XPathin käyttökohteet

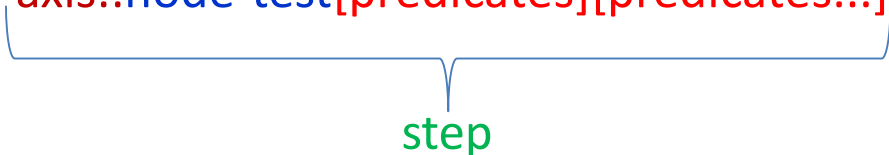
- XPath-lausekkeita käytetään laajalti hyväksi XSLT:ssa
 - XSLT on kieli XML-dokumentin muuntamiseksi muodosta toiseen
 - XSLT:ta ei voi käyttää ilman XPath-osaamista
- XPath on myös osa dokumenttien välistä linkitystä
 - XPathilla voidaan linkittää dokumentin sisälle
- XPath-lausekkeet ovat näppäriä apuvälineitä ohjelmointikielissä
 - yksinkertaisen valintalausekkeen toteuttaminen Javalla tai C:lla voi olla kymmeniä riviä koodia
 - XPath-käsittelijäkirjastot ovat valmiiksi optimoituja ja testattuja kovassa käytössä

Online xpath tool: <http://www.xpathtester.com/xpath>

XPath: rakenne

- XPath-lauseke koostuu joukosta **askelia** (step)
 - askelista muodostuu **polku** (path) alkupisteestä (current node) **loppujoukkoon**
 - askeleet erotetaan toisistaan kauttaviivoilla **/step/step/step**
 - jokainen askel muokkaa valittua joukkoa (set)
 - » jos askel päättyy tyhjään joukkoon, palautetaan tyhjä joukko
 - askel määrää **siirtymän** (axis), **solmutestin** (node test) ja mahdollisesti joukon **ehtoja** (predicates)

Rakenne: **axis::node-test**[**predicates**][**predicates...**]



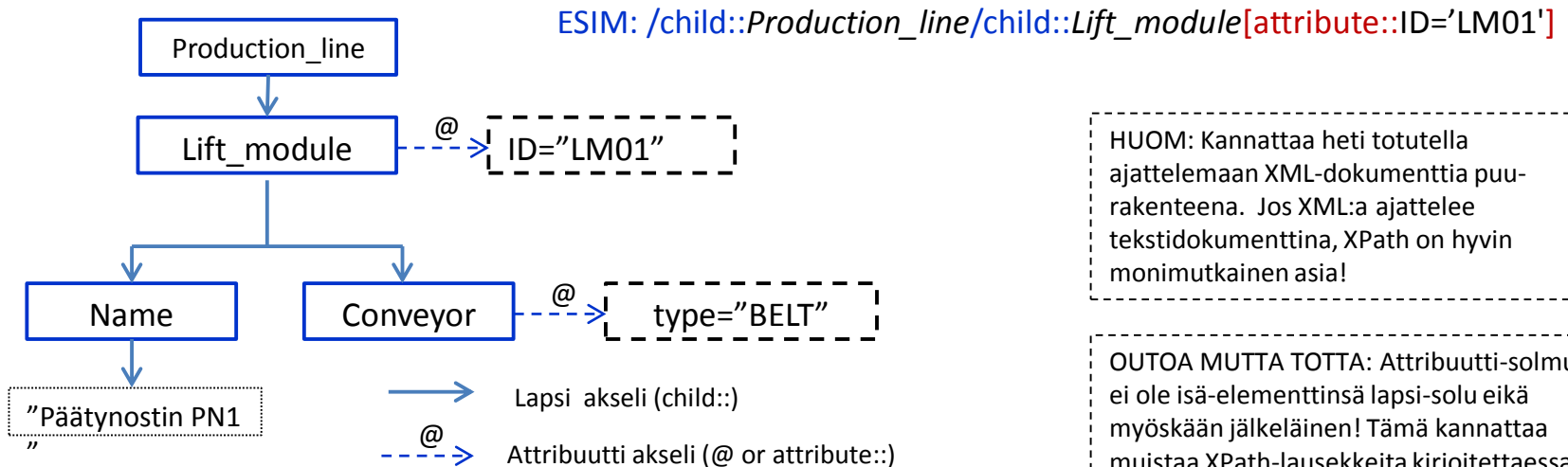
step

Esim: **child::title**[**attribute::lang='fi'**]



XPath: Lausekkeen evaluointi

- XPath-lauseke evaluoidaan aina puumuotoisessa XML:ssa
 - Elementit, attribuutit ja tekstisisällöt ovat puun solmuja (node)
- Lauseke evaluoidaan askel askeleelta
 - ensimmäinen askel otetaan esimerkiksi ns. nykyisestä elementistä (current node/ context node) tai juurielementistä
 - seuraava askel otetaan siitä, mihin edellinen jäi (se on siis uusi kontekstisolmu)
 - jos askelta ei pystytä ottamaan, lauseke palauttaa tyhjän joukon



HUOM: Kannattaa heti totutella ajattelemaan XML-dokumenttia puurakenteena. Jos XML:a ajattelee tekstidokumenttina, XPath on hyvin monimutkainen asia!

OUTOA MUTTA TOTTA: Attribuutti-solmu ei ole isä-elementtinsä lapsi-solu eikä myöskään jälkeläinen! Tämä kannattaa muistaa XPath-lausekkeita kirjoitettaessa.

XPath: Joukko-oppia

- XPath operoi joukoilla
 - jokainen askel voi saada **lähtökohdaksi useita solmuja** XML-puussa
 - solmujen tyyppien ei välttämättä tarvitse olla samoja
- Jokainen askel otetaan jokaisesta lähtöjoukon solmusta
 - mikäli askelta ei voida ottaa, kyseinen haara sulkeutuu
 - askel voi **tuottaa useita solmuja**, jotka kaikki ovat mukana seuraavassa askeleessa
- Viimeisen askeleen valitsemat solmut palautetaan XPath-lausekkeen tuloksena

XPath: Lyhennetty syntaksi

- XPath-lausekkeista on laadittu erityinen **lyhennetty syntaksi**
 - Lyhennettyä syntaksia voidaan käyttää normaalin syntaksin sijaan tai sen seassa
- Huom: kaikkea ei voida kuitenkaan toteuttaa lyhennetyllä syntaksilla
- Esimerkkejä peruslausekkeista:
 - Nykyisen elementin *title*-lapsielementtien valinta
 - `child::title` tai `title`
 - huomaa, että nykyistä elementtiä ei tarvitse merkitä erikseen, sillä XPath aloittaa oletusarvoisesti nykyisestä elementistä
 - nykyinen elementti voidaan toki merkitä alkuun
 - `self::node()/child::title` tai `./title`
 - Nykyisen elementin *src*-attribuutin valinta
 - `attribute::src` tai `@src`
 - Huomaa, että seuraavat lausekkeet palauttavat eri joukon:
 - `self::node()/child::node()` – palauttaa kaikki lapsisolmut, myös tekstisisällöt
 - `./*` – palauttaa vain lapsielementit



XPath: Askeleiden yhdistäminen

- Askeleet yhdistetään kauttaviivoilla /
 - lauseke suoritetaan askel askeleelta

`child::chapter/child::paragraph` -tai- `chapter/paragraph`

`child::paragraph/text()` -tai- `paragraph/text()`

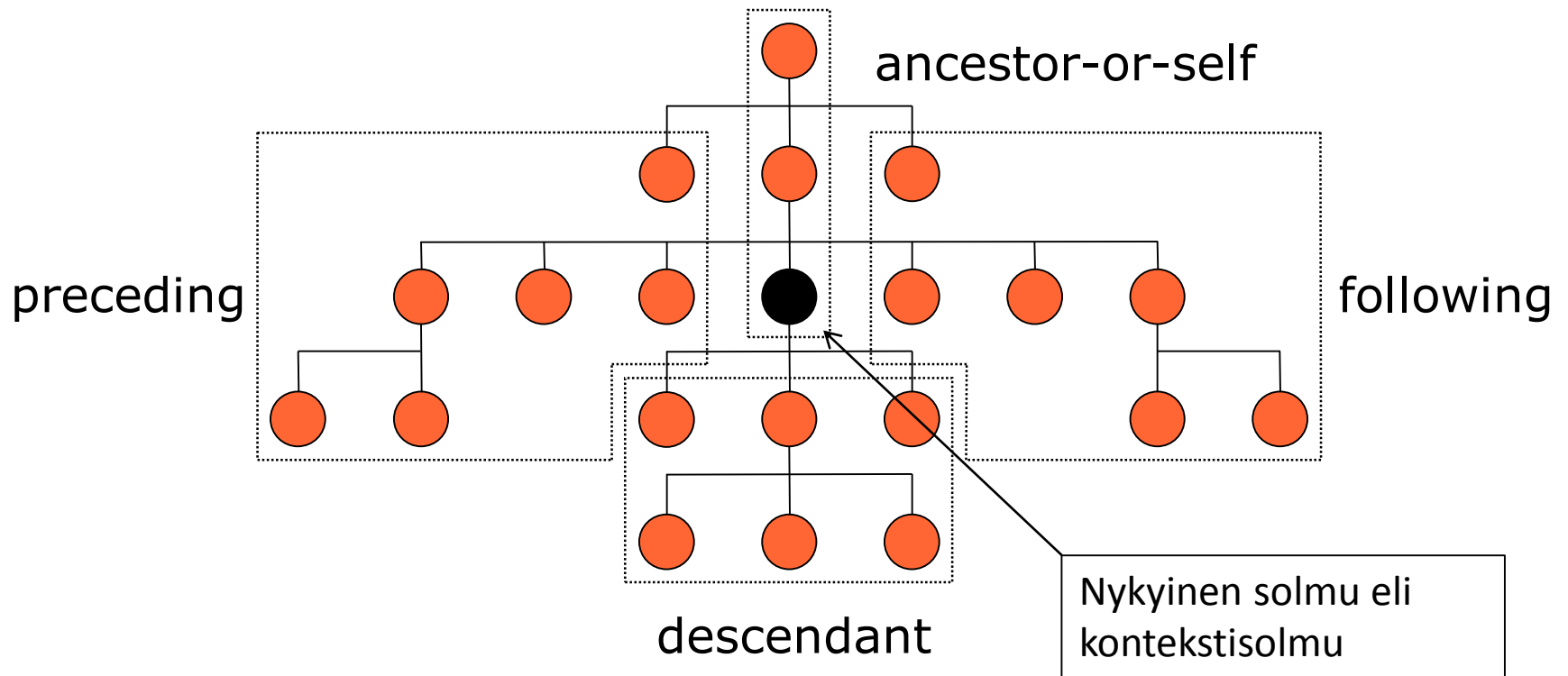
HUOM: Kauttaviiva / on sekä askelten yhdistämismerkki että lyhennysmerkintänä lapsielementtiin siirtymä askel esim: `./book`

XPath: Siirtymät XML-puussa

- XPath mahdollistaa liikkumisen kaikkiin suuntiin XML-puussa:
 - **parent** tai **..** (kaksi pistettä) valitsee isäelementin
 - **child** tai **tyhjä** valitsee lapsisolmut
 - **/** valitsee dokumentin juuren
 - **descendant** valitsee kaikki lapsielementit, myös lapsien lapset jne.
 - » **descendant-or-self** tai **//** valitsee myös nykyisen solmun
 - **ancestor** valitsee kaikki esi-isät
 - » **ancestor-or-self** valitsee myös nykyisen solmun
 - **following** ja **preceding** valitsevat kaikki dokumentissa seuraavat tai edeltävät solmut
 - **following-sibling** ja **preceding-sibling** valitsevat seuraavat tai edeltävät sisarsolmut
 - **self** tai **.** (piste) valitsee nykyisen solmun



XPath: Siirtymät esimerkki 2



Solmutesti -Siirtymien tarkentaminen

- Siirtymälle voidaan asettaa **ehtoja solmutestillä**
- Askel valitsee ainoastaan ne solmut, jotka ovat siirtymän päässä nykyisestä joukosta ja jotka täyttävät solmutestin ehdot
- **Solmutestillä** voidaan valita:
 - tietyt elementit, pelkkä leipäteksti, kommentit tai käsittelyohjeet

`following-sibling::image`

`child::text()` tai `text()`

`parent/child::comment()` tai `../comment()`

`descendant-or-self::*` tai `//*`

Ehdot

- Valintaa voidaan tarkentaa edelleen määrittelemällä haluttu määrä **ehtoja valittaville solmuille** (predicates)
 - ehdot määritetään hakasuluissa solmutestin jälkeen **[ehto]**
 - useita ehtoja voidaan määritellä peräkkäin, jokaisen on toteuduttava

`child::title[attribute::lang]` tai
`title[@lang]`

`child::title[attribute::lang='fi']` tai `title[@lang='fi']`

`ancestor-or-self::node()[child::paragraph]/child::image` tai

- `ancestor-or-self::node()[paragraph]/image`

Listing Xpath axis and node test types

XPATH:N SIIRTYMÄ- JA SOLMUTESTITYYPIT

XPath: yleisimpiä polun askeleita

- Yleisiä polkulausekkeita:

Expression	Description
<i>nodename</i>	Selects all nodes with the name " <i>nodename</i> "
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

XPath lausekkeen ensimmäisenä merkinä käytettäessä osoittaa dokumentin juureen (muuten polun askeleiden erotusmerkki).

Ex:

//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
/bookstore	Selects the root element bookstore

Note: If the path starts with a slash (/) it always represents an absolute path to an element!

XPath: siirtymäakselit

XPath Axes

An axis defines a node-set relative to the current node.

AxisName	Result
ancestor	Selects all ancestors (parent, grandparent, etc.) of the current node
ancestor-or-self	Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself
attribute	Selects all attributes of the current node
child	Selects all children of the current node
descendant	Selects all descendants (children, grandchildren, etc.) of the current node
descendant-or-self	Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself
following	Selects everything in the document after the closing tag of the current node
following-sibling	Selects all siblings after the current node
namespace	Selects all namespace nodes of the current node
parent	Selects the parent of the current node
preceding	Selects everything in the document that is before the start tag of the current node
preceding-sibling	Selects all siblings before the current node
self	Selects the current node

XPath:solmutesti (1)

TABLE 7-1: XPath Node Types and Node Tests

NODE TEST	NODE TYPES MATCHED
<code>node()</code>	Matches any node at all.
<code>text()</code>	A text node.
<code>processing-instruction()</code>	A processing instruction.
<code>comment()</code>	A comment node (if the XML processor didn't remove them from the input!).
<code>prefix:name</code>	This is called a QName, and matches any element with the same namespace URI as that to which the prefix is bound and the same "local name" as <i>name</i> (see the section "XPath and Namespaces" later in this chapter for an English translation of this!). Examples: <code>svg:circle</code> , <code>html:div</code> .
<code>name</code>	An element with the given name (entry, body, and so on).

XPath:solmutesti (2)

TABLE 7-1 (continued)

NODE TEST	NODE TYPES MATCHED
<code>@attr</code>	An attribute of the given name (id, href, and so on); can be prefixed as a QName, for example, <code>@xlink:href</code> , but remember that attributes are not in any namespace by default.
<code>*</code>	Any element.
<code>element(name, type)</code>	An element of given name (use <code>*</code> for any), and of the given schema type, for example, <code>xs:decimal</code> . Examples: <code>entry/p/element(*, xs:gYear)</code> to find any element declared with XSD to have content of type <code>xs:gYear</code> ; <code>entry/p/element(born, *)</code> , which is essentially the same as <code>entry/p/born</code> .
<code>attribute(name, type)</code>	Same as <code>element()</code> but for attributes.

XPath: askeleet

```
axisname::nodetest [predicate]
```

Examples

Example	Result
child::book	Selects all book nodes that are children of the current node
attribute::lang	Selects the lang attribute of the current node
child::*	Selects all element children of the current node
attribute::*	Selects all attributes of the current node
child::text()	Selects all text node children of the current node
child::node()	Selects all children of the current node
descendant::book	Selects all book descendants of the current node
ancestor::book	Selects all book ancestors of the current node
ancestor-or-self::book	Selects all book ancestors of the current node - and the current as well if it is a book node
child::* / child::price	Selects all price grandchildren of the current node

(*) {

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
  </book>
  <book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
  </book>
</bookstore>
```

[*Katso myös kalvo: 'solmu testi']

XPATH – OPERAATTORIT JA FUNKTIOT

www.w3schools.com - XPath Operaattorit

- XPath Operaattorit http://www.w3schools.com/xpath/xpath_operators.asp
- Lista Xpath-lausekkeissa käytettävistä operaattoreista:

Operator	Description	Example	Return value
	Computes two node-sets	//book //cd	Returns a node-set with all book and cd elements
+	Addition	6 + 4	10
-	Subtraction	6 - 4	2
*	Multiplication	6 * 4	24
div	Division	8 div 4	2
=	Equal	price=9.80	true if price is 9.80 false if price is 9.90
!=	Not equal	price!=9.80	true if price is 9.90 false if price is 9.80
<	Less than	price<9.80	true if price is 9.00 false if price is 9.80
<=	Less than or equal to	price<=9.80	true if price is 9.00 false if price is 9.90
>	Greater than	price>9.80	true if price is 9.90 false if price is 9.80
>=	Greater than or equal to	price>=9.80	true if price is 9.90 false if price is 9.70
or	or	price=9.80 or price=9.70	true if price is 9.80 false if price is 9.50
and	and	price>9.00 and price<9.90	true if price is 9.80 false if price is 8.50
mod	Modulus (division remainder)	5 mod 2	1

HUOM:
Tässä verrataan *price*-
elementin arvoa
tekstikentässä lukuun
9.80 (ei itse elementtiä)

XPath: Funktiot

- XPath määrittää myös joukon funktioita, joita voidaan käyttää lausekkeiden ehdoissa tai lausekkeiden sijaan palauttamaan jotakin tietoa dokumentista
 - funktiot ovat tyyppitettyjä
 - osa funktioista voi ottaa solmujoukon (node set) parametrikseen

XPATH: /bookstore/book[@category = 'web']/title[**string-length()** > 12]

RESULT:

<title lang="en">XQuery Kick Start</title>

XPath: Boolean-funktiot

- `boolean(obj)` muokkaa argumentin boolean-tyyppiseksi
- `false()` palauttaa aina `false`
- `not(b)` palauttaa `b`:n käänteisarvon
- `true()` palauttaa aina `true`

XPath:Numerofunktiot

- `ceiling(number)` palauttaa pienimmän mahdollisen kokonaisluvun, joka on suurempi tai yhtä suuri kuin argumentti
 - siis pyöristys ylöspäin
- `floor(number)` palauttaa suurimman mahdollisen kokonaisluvun, joka on pienempi tai yhtä suuri kuin argumentti
 - siis pyöristys alaspäin
- `number(obj)` muuttaa argumentin numeroksi
- `round(number)` pyöristää argumentin lähimpään kokonaislukuun


XPath:Merkkijonofunktiot

- `concat(s1, s2, ...)` yhdistää argumentit yhdeksi merkkijonoksi
 - argumenttien täytyy olla merkkijonoja
- `contains(haystack, needle)` palauttaa `true`, jos toinen argumentti löytyy ensimmäisestä
 - esim: `contains('XML','XM')` – palauttaa: `true`
- `normalize-space(s)` poistaa argumentin alusta ja lopusta tyhjät merkit ja korvaa kaikki tyhjien merkkien ryhmät yhdellä välilyönnillä
- `starts-with(s1, s2)` palauttaa `true`, jos `s1` alkaa `s2`:lla
- `string(obj)` muuttaa argumentin merkkijonoksi
- `string-length(s1)` palauttaa argumentin pituuden merkkeinä
- `substring(s, start, len)` palauttaa merkkijonon, joka alkaa kohdasta `start` merkkijonossa `s` ja jonka pituus on `len` merkkiä

XPath:Merkkijonofunktiot

- `substring-after(s1, s2)` palauttaa merkkijonon, joka alkaa siitä kohdin, mistä `s2` löytyy `s1`:stä
 - esim. `substring-after('Punainen', 'a')` – palauttaa: `'inen'`
- `substring-before(s1, s2)` palauttaa merkkijonon, joka päättyy siihen, mistä `s2` löytyy `s1`:stä
 - esim. `substring-before('Punainen', 'a')` – palauttaa: `'Pun'`
- `translate(s1, s2, s3)` korvaa merkkijonossa `s1` `s2`:n merkit `s3`:n merkeillä
 - esim. `translate('Punainen', 'P', 'p')` – palauttaa: `'punainen'`

Solmujoukkofunktiot

- `count(ns)` laskee solmujoukon solmujen lukumäärän
- `id(idref)` palauttaa solmun, jolla on ID-tyyppinen attribuutti, jonka arvo on `idref`
- `local-name(n)` palauttaa solmun paikallisen nimen (siis ilman nimiavaruutta) 
- `name(n)` palauttaa solmun nimen, sisältäen nimiavaruuden
- `namespace-uri(n)` palauttaa solmun nimiavaruuden URI:n
- `sum(ns)` laskee solmujoukon summan, solmut muutetaan numeroiksi

Nykyiseen solmuun liittyvät funktiot

- `position()` palauttaa nykyisen solmun järjestysnumeron nykyisessä solmujoukossa
 - ei ole sama asia kuin elementin järjestysnumero dokumentissa
 - » tätä ei ole itse asiassa mahdollista edes helposti laskea
- `last()` palauttaa nykyisen solmujoukon viimeisen solmun järjestysnumeron

XPath:Funktioiden käyttö

- Attribuuttien arvojen yhdistäminen

`concat(@firstname, ' ', @surname)`

- Vain ensimmäinen lapsielementti valitaan

`child::test[position()=1]` tai `test[position()=1]`

- Lasketaan tiettyjen elementtien lukumäärä

`count(child::paragraph) + count(child::image[attribute::src])` tai `count(paragraph) + count(image[@src])`

XPath: Funktiot - Esimerkkejä

- XPATH>/bookstore/book[@category = 'web']/title[**string-length()** > 12]
 - RESULT> <title lang="en">XQuery Kick Start</title>
- XPATH> **contains**(/bookstore/book[@category = 'web']/title, 'XQuery')
 - RESULT> true
- XPATH> //book[**contains**(./title, 'XQuery')]
 - RESULT> <book category="web">
 - <title lang="en">XQuery Kick Start</title>
 - </book>
- XPATH> //book[**substring-before**(./year/text(), '5')][./price >=30.00]/title
 - RESULT> <title lang="en">Everyday Italian</title>
- XPATH> //book[position()=1]/price/text() +
//book[position()=last()]/price/text()
 - RESULT> 69.95
- XPATH> **sum**(//book/price/text())
 - RESULT> 149.93

www.w3school.org

LISÄÄ XPATH ESIMERKKEJÄ

www.w3schools.com - XPath Examples

- **Books.xml**

- Selects the root of the document:
 - /
- Selects the bookstore root element:
 - /bookstore
- Selects all the book nodes:
 - /bookstore/child::book
 - /bookstore/book
 - //book
 - //book/.
 - /bookstore/book/year/../..//*
- Selects all the child-elements of all the books:
 - /bookstore/book/year/../../*

NOTE: Remember to use the entity values of the <, > and & characters in xpath expressions in xml documents.

http://www.w3schools.com/xpath/xpath_examples.asp

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>

<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>

<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>

</bookstore>
```

www.w3schools.com - XPath Examples

- **Books.xml**

- Selects all the title nodes:
 - `/bookstore/book/title`
 - `/bookstore/child::book/child::title`
- Selects the title of the first book node under the bookstore element:
 - `/bookstore/book[1]/title`
- Selects the texts from all the price nodes:
 - `/bookstore/book/price/text()`
- Selects all the title nodes with a price higher than 35:
 - `/bookstore/book[price>35]/title`
- Selects the parent nodes of all the English title nodes, i.e. all book nodes
 - `/bookstore/book/child::title[@lang="en"]/parent::node()`
 - **RESULT: all the book nodes**

NOTE: Remember to use the entity values of the <, > and & characters in xpath expressions in xml documents.

http://www.w3schools.com/xpath/xpath_examples.asp

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

www.w3schools.com - XPath Examples

- **Books.xml**

- Selects all the **child nodes** of the year element of the first book:

- `/bookstore/book[1]/child::year/child::node()`

- **Result set:**

- **text-node:** "2005"

- Selects all the **child elements** of the year element of the first book:

- `/bookstore/book[1]/child::year/*`

- **Result set:**

- **NO RESULT**, because `/*` selects only child elements and there is no child element.

NOTE: Remember to use the entity values of the `<`, `>` and `&` characters in xpath expressions in xml documents.

http://www.w3schools.com/xpath/xpath_examples.asp

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>

<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>

<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>

</bookstore>
```

www.w3schools.com - XPath Examples

- **Books.xml**

- Selects all descendant nodes of the first book:

- `//book[1]/descendant::node()`

- **Result set:**

- *title*-node; text-node: "Everyday Italian"
 - *author*-node; text-node: "Giada De Laurentiis"
 - *year*-node; text-node: "2005"
 - *price*-node; text-node: "30.00"

- However, attributes do not belong to this set!

OUTOA MUTTA TOTTA: Attribuutti-solmu ei ole isä-elementtinsä lapsi-solu eikä siis myöskään jälkeläinen! Tämä kannattaa muistaa XPath-lausekkeita kirjoitettaessa.

NOTE: Remember to use the entity values of the <, > and & characters in xpath expressions in xml documents.

http://www.w3schools.com/xpath/xpath_examples.asp

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<bookstore>
```

```
<book category="COOKING">
```

```
<title lang="en">Everyday Italian</title>
```

```
<author>Giada De Laurentiis</author>
```

```
<year>2005</year>
```

```
<price>30.00</price>
```

```
</book>
```

```
<book category="CHILDREN">
```

```
<title lang="en">Harry Potter</title>
```

```
<author>J K. Rowling</author>
```

```
<year>2005</year>
```

```
<price>29.99</price>
```

```
</book>
```

```
<book category="WEB">
```

```
<title lang="en">XQuery Kick Start</title>
```

```
<author>James McGovern</author>
```

```
<author>Per Bothner</author>
```

```
<author>Kurt Cagle</author>
```

```
<author>James Linn</author>
```

```
<author>Vaidyanathan Nagarajan</author>
```

```
<year>2003</year>
```

```
<price>49.99</price>
```

```
</book>
```

```
<book category="WEB">
```

```
<title lang="en">Learning XML</title>
```

```
<author>Erik T. Ray</author>
```

```
<year>2003</year>
```

```
<price>39.95</price>
```

```
</book>
```

```
</bookstore>
```

www.w3schools.com – XPath Function Examples

- **Books.xml**
- Selects all *price* or *year* elements
 - `//price | //year`
- Selects the *title* of the second last *book* node under the *bookstore* element:
 - `/bookstore/book[position()=last()-1]/title`
- Counts the number of *book* nodes:
 - `count(/bookstore/book) = 4`
- Calculates max and average values:
 - `max(//price/text()) = 49.99`
 - `avg(//price) = 37.4825`
- Selects the texts from all the *price* nodes:
 - `/bookstore/book/price/text()`
- Selects all the *title* nodes with publication *year* different from 2005:
 - `/bookstore/book[not(./year/text()='2005')]/title`

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>

<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>

<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>

</bookstore>
```

NOTE: Remember to use the entity values of the <, > and & characters in xpath expressions in xml documents.

Lähdeviitteet

- Kirjoja:
 - (XML:n perusteet ja Xpath: luvut 1, 3 ja 4 kirjasta) Goldberg, K.,H.2009. XML-Visual QuickStart Guide. 2.e. Peachpit Press XML. : [Introduction](#), [Ch1](#), Ch2, [Ch3](#), [Ch4](#) ja Ch9-Ch15.
 - (XML:n perusteet ja Xpath: luvut 1, 2 ja 7 ja Appendix B: XPath Functions kirjasta) E-book: Beginning XML, (5th ed.) by Fawcett, J., et al., John Wiley & Sons, Inc. Ch 1-3, Ch5, Ch7, Ch8.
- Muita lähteitä:
 - Tutorials: <http://www.w3schools.com/>
 - XPATH esimerkkejä: http://www.w3schools.com/xpath/xpath_examples.asp
 - Well-formed validator by w3c: http://validator.w3.org/#validate_by_input
 - W3C:n Määrittelyt:
 - <http://www.w3.org/XML/Core/>
 - <http://www.w3.org/Style/XSL/>;
 - <http://www.w3.org/XML/Schema#dev>
- Tools
 - Online tool: xpath, xslt, schema validate: <http://www.xpathtester.com>
 - Online tool: xpath, xslt, schema validate, reg-exp: <http://www.freeformatter.com>
 - Command line tool: Xmlstarlet command line XML toolkit:
 - Download for windows: <http://xmlstar.sourceforge.net/download.php>
 - XML editors
 - EditX Lite free version: <http://www.freexmleditorsite.com/download.html>
 - MindFusion XML Viewer on Ilmainen editori. Sopii ainakin XSLT-muunnoksien harjoitteluun.