

## Luento 4: XPath ja XLink

AS-0.110 XML-kuvauskielten perusteet

Janne Kalliola

### XPath ja XLink

- XPath
  - lausekkeet
  - joukko-oppi
  - askeleet
    - siirtymät
    - solmutestit
    - ehdot
  - funktiot
  - XPathin käyttö
- XLink
- XML Base
- XPointer

- 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 sitten niitä käytetään lähtökohtana erilaisille toimenpiteille, esimerkiksi XSLT-muunnoksilla (tästä myöhemmillä luennoilla)

- XPath-lauseke koostuu joukosta askelia (step)
  - askelista muodostuu polku (path) alkupisteestä loppujoukkoon
  - jokainen askel muokkaa valittua joukkoa (set)
    - jos askel päättyy tyhjäan joukkoon, palautetaan tyhjä joukko
  - askel määrää siirtymän (axis), solmutestin (node test) ja mahdollisesti joukon ehtoja (predicates)

```
axis::node-test [predicates] [predicates...]
```

- askeleet erotetaan toisistaan kauttaviivoilla

- XPath-lauseke evaluoidaan aina puumuotoisessa XML:ssa
  - jos XML:a ajattelee tekstidokumenttina, XPath on hyvin monimutkainen asia
  - kannattaa heti kättelyssä totutella ajattelemaan XML-dokumenttia puuna
    - tätä taitoa tarvitaan pitkin kurssia
- Lauseke evaluoidaan askel askeleelta
  - ensimmäinen askel otetaan esimerkiksi ns. nykyisestä (current) elementistä tai juurielementistä
    - tämä riippuu lausekkeesta
  - seuraava askel otetaan siitä, mihin edellinen jäi
  - jos askelta ei pystytä ottamaan, lauseke palauttaa tyhjän joukon

- 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-lausekkeista tulee helposti pitkiä ja hankalia lukea
- Tämän vuoksi on laadittu erityinen lyhennetty syntaksi, jossa kaikkein oleellisimmat siirtymät on lyhennetty parin merkin mittaisiksi
- Lyhennettyä syntaksia voidaan käyttää normaalin syntaksin sijaan tai sen seassa
  - kaikkea ei voida kuitenkaan toteuttaa lyhennetyllä syntaksilla
- Tällä kurssilla käydään molemmat syntaksit läpi

- Nykyisen elementin title-lapsielementtien valinta

```
child::title tai title
```

- huomaa, että nykyistä elementtiä ei tarvitse merkitä erikseen
- XPath aloittaa oletusarvoisesti nykyisestä elementistä
  - nykyinen elementti voidaan toki merkitä alkuun

```
self/child::title tai ./title
```

- Nykyisen elementin src-attribuutin valinta

```
attribute::src tai @src
```

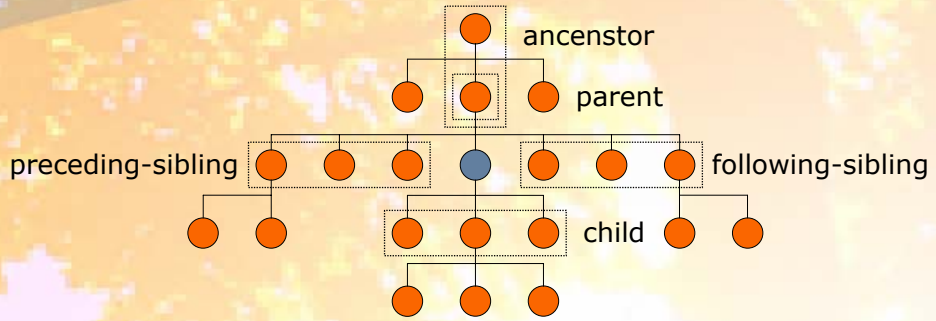
- Askeleet yhdistetään kauttaviivoilla
  - lauseke suoritetaan askel askeleelta

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

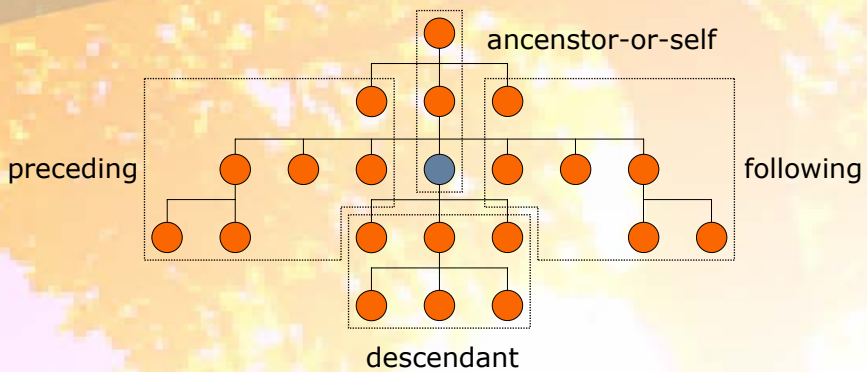
- XPath mahdollistaa liikkumisen kaikkiin suuntiin XML-puussa
  - `parent` **tai** `..` valitsee isäelementin
  - `child` **tai** tyhjä valitsee lapsielementit
  - `/` valitsee juurielementtiin
  - `descendant` **tai** `//` valitsee kaikki lapsielementit, myös lapsien lapset jne.
    - `descendant-or-self` 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
    - toisin sanoen kaikki elementit, jotka ovat vielä avautumatta (`following`) tai ovat jo sulkeutuneet (`preceding`)
  - `following-sibling` **ja** `preceding-sibling` valitsevat seuraavat tai edeltävät sisarsolmut
  - `self` **tai** `.` valitsee nykyisen solmun



## Siirtymät, esimerkki (1/2)



## Siirtymät, esimerkki (2/2)



- 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::* tai /**
```

- Valintaa voidaan tarkentaa edelleen määrittelemällä haluttu määrä ehtoja valittaville solmuille
  - ehdot määritetään hakasuluissa solmutestin jälkeen
  - 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[child::paragraph]/  
child:image tai ancestor-or-  
self[paragraph]/image
```

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

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



- `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

- `concat(s1, s2, ...)` yhdistää argumentit yhdeksi merkkijonoksi
  - argumenttien täytyy olla merkkijonoja
- `contains(haystack, needle)` palauttaa true, jos toinen argumentti löytyy ensimmäisestä
- `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

- `substring-after(s1, s2)` palauttaa merkkijonon, joka alkaa siitä kohdin, mistä `s2` löytyy `s1`:stä
  - esimerkiksi `substring-after('XPath', 'P')` palauttaa `'ath'`
- `substring-before(s1, s2)` palauttaa merkkijonon, joka päättyy siihen, mistä `s2` löytyy `s1`:stä
  - esimerkiksi `substring-before('XPath', 'P')` palauttaa `'X'`
- `translate(s1, s2, s3)` korvaa merkkijonossa `s1` `s2`:n merkit `s3`:n merkeillä
  - esimerkiksi `translate('XPath', 'XP', 'xp')` palauttaa `'xpath'`

- `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-url(n)` palauttaa solmun nimiavaruuden URI:n
- `sum(ns)` laskee solmujoukon summan, solmut muutetaan numeroiksi

- `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

- 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-lausekkeita käytetään laajalti hyväksi XSLT:ssa
  - XSLT on kieli XML-dokumentin muuntamiseksi muodosta toiseen
  - XSLT:ta ei voi käyttää ilman XPath-osaamista
- XPath on osa myös 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

## XLink

- XLink on kokoelma standardeja, joilla mahdollistetaan hyperlinkit XML-dokumenteista
  - hyperlinkit käsitetään tässä yhteydessä laajemmin kuin HTML:n yksisuuntaiset yhdestä-yhteen linkit
  - perusteet ovat silti samat
    - resurssiin viitataan URI:lla

- Resurssi – URI:lla viitattava tietosisältö
- Linkki – viittaus kahden resurssin välillä
- Hyperlinkki – pääasiassa ihmisen käyttöön suunnattu linkki
- seuraaminen (traversal) – siirtyminen linkin avulla resurssista toiseen



- XLink ei määrittele mitään erityistä XML-kieltä, vaan tarjoaa tietyn joukon käsitteitä ja attribuutteja käytettäväksi missä tahansa XML-pohjaisessa kielessä
  - XLinkin attribuutit sijaitsevat omassa nimiavaruudessa, jonka URI on <http://www.w3.org/1999/xlink>
    - lyhennetään yleensä xlink
  - näillä attribuuteilla voidaan määrittää linkin tyyppi, viitattava resurssi ja antaa ohjeita dokumenttia käsittelevälle ohjelmalle
  - kaikkia attribuutteja ei ole pakko käyttää

```
<my:crossReference
  xmlns:my="http://example.com/"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="simple"
  xlink:href="students.xml"
  xlink:role="http://www.example.com/linkprop
s/studentlist"
  xlink:title="Student List"
  xlink:show="new"
  xlink:actuate="onRequest">
  Current List of Students
</my:crossReference>
```

- XLink määrittää kaksi erilaista linkkityyppiä
  - yksinkertainen linkki (simple link) on viittaus yhdestä resurssista toiseen
    - kaikki HTML:n linkit ovat tätä tyyppiä
  - laajennettu linkki (extended link) on monimutkainen viittausrakennelma
    - mahdollisuus viitata useaan dokumenttiin
    - kaksisuuntaiset linkit ovat sallittuja
    - linkin seuraamiselle voidaan asettaa sääntöjä
- Kurssin kannalta yksinkertainen linkkityyppi on riittävä

## XML Base

- XML Basen avulla voidaan määrittää dokumentille perus-URI
  - vastaa HTML:n base-elementtiä
  - kaikki suhteelliset viittaukset dokumentista alkavat perus-URI:sta
  - absoluuttiset viittaukset toimivat kuten ennenkin
- XML Base määrittää yhden attribuutin *xml:base*, joka voidaan sijoittaa XML-dokumenttiin perus-URI:n määrittämiseksi

```
<doc xml:base="http://example.org/today/"  
  xmlns:xlink="http://www.w3.org/1999/xlink">  
  <head>  
    ...  
  </head>  
</doc>
```

## XPointer

- XPointer on linkitykseen liittyvä tapa viitata XML-dokumentin sisällä oleviin elementteihin tai muihin solmuihin
  - vastaa eräällä tavalla HTML:n #anchor –linkitystä
- Viittaukset voidaan tehdä muun muassa elementtien järjestyksen tai nimiavaruuksien perusteella
  - myös XPath on mahdollinen vaihtoehto

- XPointer-viittaukset perustuvat skeemoihin
  - xpointer() –skeema
    - mahdollistaa XPath-lausekkeiden suorittamisen XPointer-viittauksena
    - määrittää joukon omia funktioita
  - element() –skeema
    - elementteihin viitataan järjestysnumerolla
    - viittaus aloitetaan joko juurielementistä tai id-attribuutin perusteella
    - esimerkiksi:

```
element(intro)
element(/1/3/5)
element(intro/2)
```

**Kysymyksiä? Kommentteja?**