

Luento 11: XSL-FO & SVG

AS-0.110 XML-kuvauskielten perusteet

Janne Kalliola

XSL-FO & SVG

- XSL-FO
 - Dokumentin rakenne
 - Sivupohja ja sivujaksopohja
 - Sisältö
 - Muut ominaisuudet
 - Dokumentin laadinta ja käyttö
 - XSL-FO:n ongelmat
- SVG
 - Scalable Vector Graphics
 - SVG-dokumentti
 - Grafiikkaoliot
 - symbolit
 - ryhmittely
 - Dynaamisuus
 - SVG:n tuottaminen ja käyttö

- XSL Formatting Objects (XSL-FO) on kieli ulkoasun kuvaamiseksi
 - kielen avulla voidaan luoda esimerkiksi tulostettavia dokumentteja
 - kieli sisältää toimintoja myös online-julkaisuun ja tekstin ääneen lukemiseen
 - pohjautuu voimakkaasti Cascading Stylesheets (CSS) versioon 2
 - lisää ja muokkaa CSS:n määrittelyä
- Kieli on sangen monimutkainen
 - ei kannata kirjoittaa käsin - ainakaan suuria määriä
 - yleensä XSL-FO-dokumentti syntyy XSLT-tyylisivun avulla
 - on olemassa myös XSL-FO-pohjaisia ohjelmia
 - valitettavan aikaisessa kehitysvaiheessa tosin
- Tällä luennolla ei käsitellä kaikkia FO:n piirteitä
 - koodiesimerkeissä ei välttämättä ole kaikkea tarpeellista

- FO-dokumentti koostuu
 - pohjien määrittelyistä
 - sivupohjista (page master)
 - sivujaksopohjista (page sequence master)
 - sivujaksoista (page sequence)
 - Dokumentin kaikki elementit ovat elementin `<fo:root>` sisällä
 - kaikki FO:n elementit ovat nimiavaruudessa <http://www.w3.org/1999/XSL/Transform>
 - nimiavaruus lyhennetään yleensä fo:

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
...
</fo:root>
```

Sivupohja

- Sivupohja määrittelee dokumentissa käytettävän yksittäisen sivun koon
 - samalla sivupohjaan määritellään käytössä olevat alueet (region)
 - alueita on maksimissaan viisi (ylös, alas, vasen, oikea, keskellä)
 - alueiden sisältö määritellään myöhemmin
 - toisin kuin tekstinkäsittely- ja DTP-ohjelmissa, FO:n sivupohjissa ei määritellä mitään näkyvää
 - sivupohjia voi olla useita ja ne voivat olla erikokoisiakin

Sivupohjaesimerkki

- Sivupohja määritellään seuraavalla elementillä

```
<fo:simple-page-master master-name="body"
  page-height="29.7cm" page-width="21cm"
  margin-top="2cm" margin-bottom="2cm"
  margin-left="2cm" margin-right="2cm">
  <fo:region-after extent="2cm" region-name="footer"/>
  <fo:region-body margin-top="1cm"
    margin-bottom="1cm"/>
</fo:simple-page-master>
```

Sivujaksopohja

- Sivujaksopohja määrittelee käytettävien sivupohjien järjestyksen
 - vähintään yhtä sivupohjaa täytyy käyttää
 - sivupohjien käytölle voidaan asettaa ehtoja
 - ensimmäinen tai viimeinen
 - parillinen tai pariton
 - tyhjä
- Dokumentti sivutetaan sivujaksopohjan perusteella
 - sivujaksopohjia on useita, alla yksinkertaisin

```
<fo:page-sequence-master master-name="bodyseq">
  <fo:single-page-master-reference master-reference="body"/>
</fo:page-sequence-master>
```

Sisältö

- FO-dokumentin sisältö sijoitetaan sivujakson sisään
 - sivujaksossa voi olla useita staattisia lohkoja (static-content) ja yksi vuo (flow)
 - sivujaksoja voi olla useita ja ne voivat käyttää eri sivujaksopohjia
 - staattiset lohkot toistuvat sivusta toiseen samoina
 - ylä- ja alaotsakkeet, sivunumerointi yms.
 - vuo sisältää leipätekstin

```
<fo:page-sequence master-reference="bodysec">
  <fo:static-content flow-name="footer">
    ..
  </fo:static-content>
  <fo:flow flow-name="xsl-region-body">
    ...
  </fo:flow>
</fo:page-sequence>
```

Sisällön rakenne

- Sisältö koostuu sisäkkäisistä ja peräkkäisistä lohkoista (block)
 - lohkojen välissä on aina rivinvaihto
 - lohkot eivät voi olla rinnakkaisia ilman erikoisrakenteita
 - taulukot (table) ja listat (list-block)
- Lohkon attribuuteilla säädellään lohkossa olevan tekstin ulkonäköä, lohkon ympärillä olevaa tyhjää tilaa ja muita vastaavia asioita

```
<fo:block font-family="Times" font-size="10pt" line-height="12pt"
space-before.optimum="6pt" space-after.optimum="12pt">
```

...

```
</fo:block>
```

Lohkot

- Lohkon koko määräytyy sisällön perusteella
 - yleensä joko lohkon leveys tai korkeus määräytyy lohkon ympärillä olevista elementeistä
 - toinen mitta syntyy sisällön taittamisella lohkon sisään
- Lohkojen etäisyyttä toisistaan voidaan säädellä
 - jokainen suunta erikseen tai kaikki yhdessä
 - lohkolle voidaan määrittää samalla myös kehys
- Lohko voi olla tyhjä
 - aiheuttaa rivinvaihdon

Rivin sisäiset rakenteet

- Lohkojen sisällä tekstin tyyliä voidaan muokata rivin sisäisillä rakenteilla
 - korostukset yms.
 - rivin sisäiset rakenteet eivät aiheuta rivinvaihtoa
`<fo:inline font-weight="bold">...</fo:inline>`
- Edellä mainitun rakenteen lisäksi osa muistakin XSL-FO:n elementeistä on rivin sisäisiä
 - kuvat
 - viivat (leader)
- Lohko voidaan myös sijoittaa rivin sisälle elementillä
`<fo:block-container>`

Attribuuttien perittävyys

- FO-dokumenteissa lähes kaikki asiat määritellään attribuuteilla
 - attribuuttien nimet ovat selkeitä ja siksi pitkiä
 - yleensä taitetussa materiaalissa sivut ja kappaleet muistuttavat toisiaan
- FO-dokumentin yksinkertaistamiseksi suurin osa attribuuteista on periytyviä
 - attribuutin arvo siirtyy automaattisesti ulommalta elementiltä sisemmälle
 - ellei sisempi määritä attribuuttia uudestaan
 - kätevä tapa määrittää esimerkiksi koko dokumentissa käytettävä kirjasin suoraan juurielementissä
 - mahdolliset poikkeukset käsitellään erikseen

Monikielisyys

- XSL-FO on suunniteltu kansainväliseksi standardiksi
 - merkistö pohjautuu Unicodeen
 - eri kielten aakkosia varten on erilaisia määrittämiä
 - eri kielillä on erilaisia kirjoitussuuntia
 - sekä rivin että sivun tasolla
 - on mahdollista vaihtaa kirjoitussuunta esimerkiksi yhden sanan ajaksi
 - XSL-FO:ssa sivun rakentaminen on eriytetty geometrisista määreistä
 - sen sijaan puhutaan käsitteillä before, after (top, bottom) ja start, end (left, right)
 - vaihdettaessa kirjoitussuuntaa käsitteet vaihtuvat vastaavasti
 - itse XSL-FO-dokumentin rakennetta (tai sitä tuottavaa XSLT-tyylisivua) ei tarvitse muokata

Muut ominaisuudet

- XSL-FO tukee aiempien ominaisuuksien lisäksi
 - leijuvia elementtejä (`fo:float`)
 - alaviitteitä (`fo:footnote`)
 - dynaamisia efektejä
 - online-julkaisuun
 - ristiinviittauksia
 - väriprofiileja (`fo:color-profile`)
 - linkkejä (`fo:basic-link`)
 - ulkoiset ja sisäiset
 - ääniominaisuuksia
 - näkövammaisille

XSL-FO-dokumentin laadinta

- Dokumentti laaditaan yleensä XSLT-tyylisivulla
 - tyylisivun rakenteen määrittää alkuperäinen dokumentti
 - alkuperäistä dokumenttia käsitteleviin mallineisiin sisällytetään fo-koodi
 - myös määrytykset
 - yleensä määrytykset liitetään alkuperäisen dokumentin juurielementin mallineeseen
 - vaarana tyylisivun ylenmääräinen monimutkaistuminen
- Muista, että XSL-FO on pelkästään ulkonäköä kuvaava kieli
 - ei ole suurta väliä kuinka joku asia toteutetaan, kunhan lopputulos näyttää halutulta
 - standardi on laaja, on useita erilaisia tapoja toteuttaa asioita

Dokumentin käyttö

- XSL-FO dokumentti näytetään tai tulostetaan erillisellä XSL-FO-ohjelmalla
 - taittaa dokumentin XSL-FO-dokumentin ohjeiden mukaan
 - sivutus
 - tekstin ladonta
 - kuvien liittäminen
 - lopputulos voidaan muuttaa myös PDF:ksi
- Suurin osa nykyisistä ohjelmista ei ole graafisia
 - komentorivityökaluja XSL-FO-dokumentin muuttamiseksi PDF:ksi

XSL-FO:n ongelmat

- XSL-FO on monimutkainen kieli
 - layout-määrittelyjen kuvaaminen ei ole helppoa
 - visualisointi hyvin vaikeaa
 - standardi on laaja
 - elementtien ja attribuuttien nimet ovat pitkiä
- XSL-FO:n tuki ei ole täydellistä yhdessäkään ohjelmassa
- Standardi ei ratkaise kaikkia ongelmia
 - esimerkiksi indeksien numerointiin voi ilmestyä tuplia
- XSL-FO ei sisällä erityistukea millekään tuotettavalla muodolle
 - PDF:lle olisi syytä olla lisätoimintoja
 - kirjanmerkit
- Osaan näistä ongelmista on ratkaisut työn alla olevassa XSL-FO 1.1 -suosituksessa

SVG

SVG

- Scalable Vector Graphics (SVG) on XML-pohjainen kieli vektorigrafiikan esittämiseen
 - tällä hetkellä käytössä versio 1.1
 - W3C määrittää versiota 1.2
 - SVG:sta on olemassa supistetut versiot langattomaan käyttöön
- SVG:lla pystyy esittämään kaikki oleelliset kaupallisista vektoripiirto-ohjelmista tutut ominaisuudet

SVG-dokumentti

- Dokumentin kaikki elementit ovat elementin `<svg:svg>` sisällä
 - kaikki SVG:n elementit ovat nimiavaruudessa <http://www.w3.org/2000/svg>
 - nimiavaruus lyhennetään yleensä `svg`:

```
<svf:svg xmlns:svg=" http://www.w3.org/2000/svg ">
```


...

```
</svg:svg>
```
 - Kuvan koko yms. määritellään juurielementissä
 - juurielementin lapsielementit voivat olla suoraan piirtoprimitiivejä
 - piirtoprimitiivien lisäksi dokumentti voi sisältää määrittämiä
 - määrittämiin voidaan viitata myöhemmin dokumentista
 - eräänlainen grafiikkakirjasto

Grafiikkaoliot

- SVG tukee kolmen tyyppisiä olioita
 - vektoripohjaiset muodot (vector graphic shapes)
 - koostuvat kaarien ja suorien viivojen muodostamista käyristä
 - perusmuodot ovat nelikulmio (rect), ympyrä (circle), soikio (ellipsi), viiva (line), moniviiva (polyline) ja monikulmio (polygon)
 - bittikuvat (images)
 - teksti
- Kaikki grafiikkaoliot sijaitsevat kanvaasilla (canvas)
- Olioiden ulkonäköä säädelään Cascading Stylesheet (CSS)-määrittelyksillä
 - CSS-koodi voidaan määritellä oliossa tai sen ulkopuolella
 - olioihin voidaan liittää myös monimutkaisempia ulkonäköasetuksia
 - liu'ut (gradient), maskit (mask)
 - asetuksiin viitataan URI:lla, voivat olla dokumentin sisäisiäkin

Symbolit

- Symboli (`svg:symbol`) on kertaalleen määritelty kokoelma grafiikkaolioita
 - symbolia voidaan käyttää (`svg:use`) uudestaan ja uudestaan
 - symbolin kokoa ja muita ominaisuuksia voidaan säätää
 - `svg:use` voi määrittää muunnoksia (transform) symbolin muokkaamiseksi
- SVG sallii myös viittaamiseen yksittäiseen grafiikkaolioon symbolin tapaan
 - olio määritelty määrittelyosiossa

Ryhmittely

- Grafiikkaolioita voidaan sisällyttää ryhmiin
 - osa olioiden asetuksista voidaan määritellä ryhmän tasolla
 - ryhmän olioihin voidaan viitata ryhmän id:lla
 - ryhmät voivat olla sisäkkäisiä
- Ryhmä piirretään
 - symbolia ei piirretä ennen sen käyttöä
 - ryhmää ei voida käyttää useita kertoja

Dynaamisuus

- SVG mahdollistaa dynaamiset kuvat
 - animaatiot
 - interaktiivisuus
- SVG-kuvaa voi muokata skriptikielellä (JavaScript)
 - käytetään Document Object Model (DOM) -rajapintaa
 - apuna lisäksi tapahtumakäsittelijät (event handlers)
 - laukaisevat koodisirpaleita tapahtumien pohjalta
 - esimerkiksi hiiren siirto tietyn grafiikkaolion päälle
- SVG-dokumentissa voi olla ehdollisia osioita
 - osiot näytetään vain, jos SVG-näyttöohjelma kykenee tiettyihin toimintoihin
 - samanlainen testi voidaan tehdä käyttäjän valitseman kielen pohjalta

SVG:n tuottaminen

- SVG-dokumentteja ei yleensä kannata laatia käsin
 - hankalaa, hidasta ja virheherkkää
- Tuottaminen XSLT:lla soveltuu toistuvaan tuottamiseen
 - esimerkiksi erilaiset käyrät
 - XSLT-tyylisivu pitää silti tehdä käsin
 - kannattaa varata aikaa
- Monet piirto-ohjelmat osaavat tallentaa SVG-muodossa
 - osa osaa myös lukea SVG-tiedostoja
 - esimerkiksi Adobe Illustrator 10, CorelDraw 11
- On olemassa erillisiä SVG-pohjaisia piirto-ohjelmia
 - yleensä Open Source -ohjelmia
 - ovat vielä lapsenkengissään

SVG:n käyttö

- SVG-kuvat vaativat tällä hetkellä erillisen näyttöohjelman
 - Adobe ja Corel tarjoavat ohjelmia ilmaiseksi
 - Javaan löytyy SVG-kirjasto
- Osa uudemmissa selaimista sisältää SVG-tuen
 - Mozilla
- SVG:ta voidaan yleensä käyttää XSL-FO:n sisällä kuvina
 - tuki riippuu XSL-FO-ohjelmasta
 - standardi ei sinänsä ota kantaa
 - SVG-kuva voi olla erillisessä tiedostossa tai XSL-FO-dokumentin sisällä

Yhteenveto

- XSL Formatting Objects (XSL-FO) on kieli ulkoasun kuvaamiseksi
 - monimutkainen alue, monimutkainen kieli
 - vaatii pitkäjänteisyyttä
 - ohjelmistotuki ei vielä riittävän pitkällä
- Scalable Vector Graphics (SVG) on kieli vektorigrafiikan esittämiseen

Kysymyksiä? Kommentteja?