Gagnasafnsfræði

Páll Melsted

11. nóv

XML

XML er staðall fyrir hálfformuð gögn sem táknar netið á línulegu formi.

- Tög í XML eru afmörkuð með `<...>` eins og í HTML (HTML er ekki afbrigði af XML).
- Tög eru opnuð með `<Foo>` og lokað með `</Foo>` allt sem er á milli er kallað stak (element)
- Sum tög hafa engin stök og því hægt að skrifa `<Foo/>`
- Tög geta haft eiginleika (attribute) skrifaðir sem `<Foo attribute="value" ... >`

Löglegt XML

XML getur verið löglegt eitt og sér (well-formed XML) eða verið löglegt miðað við skema (valid XML). Skema fyrir XML skilgreinir hvaða tög eru leyfð og hvernig þau mega hreiðrast.

Löglegt XML

```
<?xml version="1.0" encoding="utf-8" ?>
<Foo>
 ...
</Foo>
```

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<Foo>
 ...
</Foo>
```

- `<?xml ... ?>` gefur upplýsingar um XML skjalið að
  - það er XML skv. útgáfu 1.0
  - textakóðinn sem er notaður er utf-8.
- Aðeins eitt rótartag
- Öll tög sem eru opnuð þurfa að lokast
- Áður en tagi er lokað þurfa öll tög hreiðruð innni að vera lokuð (eins og lögleg svigasetning).

```
<?xml version="1.0" encoding="utf-8" ?>
<StarMovieData>
    <Star>
        <Name>Carrie Fisher</Name>
        <Address>
            <Street>123 Maple St.</Street>
            <City>Hollywood</City>
        </Address>
        <Address> ... </Address>
    </Star>
    <Star>
        <Name>Mark Hamill</Name>
        <Address>
            <Street>456 Oak Rd.</Street>
            <City>Brentwood</City>
        </Address>
    </Star>
    <Movie>
        <Title>Star Wars</Title>
        <Year>1977</Year>
    </Movie>
</StarMovieData>
```

Eiginleikar

Við getum sett upplýsingar í laufhnúta eða sem eiginleika (attribute).

Movie hefur Year og Title. Við gætum t.d. gert

```
<Movie>
    <Title>Star Wars</Title>
    <Year>1977</Year>
</Movie>
```

eða

```
<Movie year="1977">
    <Title>Star Wars</Title>
</Movie>
```

eða jafnvel

```
<Movie year="1977" title="Star Wars"/>
```

Röð eiginleika skiptir ekki máli en innbyrðis röð taga í sömu hreiðrun skiptir máli.

Eiginleikar

Helsti munurinn er að við höfum meira frelsi í laufhnútum en notum eiginleika fyrir lykla, ytri lykla og valkvæða eiginleika (optional).

Til að tengja þvert yfir tréð, eins og í starsIn, getum við notað eiginleika. Einn eiginleiki verður lykill, yfirleitt kallaður `id`, og annar vísar til þessa `id`.

```
<Star id="cf" starsIn="sw">
    <Name>Carrie Fisher</Name>
</Star>
<Star id="mh" starsIn="sw">
    <Name>Mark Hamill</Name>
</Star>
<Movie id="sw" starsOf="mh,cf">
    <Title>Star Wars</Title>
    <Year>1977</Year>
</Movie>
```

. . .

Þetta er samt frekar klunnalegt

Eiginleikar og ytri lyklar

Skárri lausn

```
<Star id="cf" starsIn="sw">
    <Name>Carrie Fisher</Name>
</Star>
<Star id="mh" starsIn="sw">
    <Name>Mark Hamill</Name>
</Star>
<Movie id="sw">
    <Title>Star Wars</Title>
    <Year>1977</Year>
    <Roles>
        <starRef>mh</starRef>
        <starRef>cf</starRef>
    </Roles>
</Movie>
```

Eins væri hægt að telja upp hlutverk fyrir hverja stjörnu.

Ytri lyklar

Hvernig er hægt að tryggja að XML skjal sé á fyrirfram ákveðnu formi?

DTD - Document Type Definition er staðall til að segja til um skema fyrir XML skjal. Gegnir sama hlutverki og `CREATE TABLE` í SQL.

DTD er með ytri lykla en er ekki nógu öflugt til að segja til um hvert þeir eiga að vísa.

XSD - XML Schema Definition leyfir sterkari tengingar.

XML namespace

Til að vísa á tag í öðru samhengi með sama nafn er hægt að nota xml namespace til að greina á milli. Tagið er þá með `xmlns` eiginleika á forminu

`xmlns:name="URI"`

þar sem `name` er nafnið á namespace og `URI` er tilvísun á skemað sem á að nota.

```
<md:StarMovieData xmlns:md="http://infolab.stanford.edu/movies">
  ...
</md:StarMovieData>
```

DTD

DTD - Document Type Definition er mállýsing fyrir XML skjöl.

```
<!DOCTYPE root-tag [
    <!ELEMENT name (components)>
    ...
]>
```

Fyrsta skilgreiningin er á rótartaginu og innan þess eru þau tög, `<!ELEMENT .. >` sem geta komið fyrir.

`components` þátturinn inniheldur önnur tög eða

- `(#PCDATA)` - texti
- `(#CDATA)` - texti sem er *ekki* túlkaður af XML
- `EMPTY` - tómt, ath. engir svigar

DTD

```
<!DOCTYPE Stars [
    <!ELEMENT  Stars (Star*)>
    <!ELEMENT  Star (Name, Address+, Movies)>
    <!ELEMENT  Name (#PCDATA)>
    <!ELEMENT  Address (Street, (City|Zip))>
```

```
    <!ELEMENT  Street (#PCDATA)>
    <!ELEMENT  City (#PCDATA)>
    <!ELEMENT  Zip (#PCDATA)>
    <!ELEMENT  Movies (Movie*)>
    <!ELEMENT  Movie (Title, year)>
    <!ELEMENT  Title (#PCDATA)>
    <!ELEMENT  Year (#PCDATA)>
]>
```

- `*` 0 eða fleiri endurtekningar
- `+` þýðir 1 eða fleiri.
- `?` 0 eða 1
- `|` velur á milli möguleika Að öðru leyti þarf hvert tag að koma nákvæmlega einu sinni og í sömu röð og er talið upp.

DTD og XML

Til að merkja skjal sem á að uppfylla DTD er sett

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<!DOCTYPE Stars SYSTEM "stars.dtd">
```

eða

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

DTD og eiginleikar

Við getum skilgreint eiginleika sem geta/verða að vera til staðar í tagi.

```
<!ATTLIST element attribute type>
```

Type getur verið `CDATA` (texti) eða upptalning á ákveðnum gildum.

```
<!ELEMENT Movie EMPTY>
    <!ATTLIST Movie
        title CDATA #REQUIRED
        year CDATA #REQUIRED
        genre (comedy | drama | sciFi) #IMPLIED
    >
```

`#REQUIRED` er fyrir eiginleika sem þurfa að vera og `#IMPLIED` fyrir valkvæða.

XML vinnsla

Til að vinna með XMl skjöl er hægt að nota nokkrar aðferðir

- DOM (Document Object Model) les XML skjal sem tré og veitir aðgang að gagnagrind
- SAX (Simple API for XML) streymir í gegnum XML skjal
- XPath er fyrirspurnarmál til að velja hnúta í XML skjali
- XQuery er fyrirspurnar- og forritunarmál til að vinna með og breyta XML skjölum
- XSLT er notað til að breyta einu XML skjali í annað

DOM

Í Java er hægt að nota `javax.xml.parsers.*`. Í python eru nokkrir valmöguleikar fyrir pakka `xml.parsers.expat`, `xml.dom.minidom` og `xml.etree.ElementTree`.

Fyrir `xml.etree.ElementTree` notum við nokkrar aðferðir

- `ET.parse()` les XML skjal og skilar tré
- `tree.getroot()` skilar rótarhnútinum
- `element.attrib` er tafla (dict) fyrir eiginleika tags
- `element.getChildren` skilar lista yfir börn
- `element.text` fyrir texta

SAX

SAX er hraðvirkari en DOM því tréð er ekki búið til og notar mun minna minni. Hins vegar þurfum við að sjá um alla vinnslu sjálf.

Við skilgreinum `ContendHandler` sem fær skilaboð um tög sem byrja, enda og texta.

- Þurfum að erfa frá `xml.sax.ContendHandler`
- Yfirskrifum `init`, `startElement`, `endElement`, `characters`
- `xml.sax.parse` sér um að kalla á viðeigandi aðferðir.

SAX

```
import xml.sax,sys

class TitleAbstractContentHandler(xml.sax.ContentHandler):
    def __init__(self):
        xml.sax.ContentHandler.__init__(self)
        self.on = False

    def startElement(self, name, attrs):
        if name == 'ArticleTitle':
            self.on = True

    def endElement(self, name):
        if name == 'ArticleTitle':
            self.on = False

    def characters(self, content):
        if self.on:
            print content

if __name__ == '__main__':
    xml.sax.parse(open(sys.argv[1]), TitleAbstractContentHandler())
```

DOM

Í Java er hægt að nota `javax.xml.parsers.*`. Í python eru nokkrir valmöguleikar fyrir pakka `xml.parsers.expat`, `xml.dom.minidom` og `xml.etree.ElementTree`.

Fyrir `xml.etree.ElementTree` notum við nokkrar aðferðir

- `ET.parse()` les XML skjal og skilar tré
- `tree.getroot()` skilar rótarhnútinum
- `element.attrib` er tafla (dict) fyrir eiginleika tags
- `list(element)` skilar lista yfir börn
- `element.text` fyrir texta

XPath

`find(path)` og `findall(path)` finna fyrsta og alla hnúta sem uppfylla ákveðin skilyrði `path`. `path` getur verið nafn á tagi eða XPath fyrirspurn

- `*` velur allt
- `.` velur hnútinn sem er valinn
- `..` velur foreldri
- `//` velur alla afkomendur
- `[1]` velur fyrsta afkomanda
- '[@attrib]' velur þá sem hafa attrib

XPath staðallinn leyfir flóknari fyrirspurnir

SAX

SAX er hraðvirkari en DOM því tréð er ekki búið til og notar mun minna minni. Hins vegar þurfum við að sjá um alla vinnslu sjálf.

Við skilgreinum `ContendHandler` sem fær skilaboð um tög sem byrja, enda og texta.

- Þurfum að erfa frá `xml.sax.ContendHandler`
- Yfirskrifum `init`, `startElement`, `endElement`, `characters`
- `xml.sax.parse` sér um að kalla á viðeigandi aðferðir.

SAX

```
import xml.sax,sys

class TitleAbstractContentHandler(xml.sax.ContentHandler):
    def __init__(self):
        xml.sax.ContentHandler.__init__(self)
        self.on = False

    def startElement(self, name, attrs):
        if name == 'ArticleTitle':
            self.on = True

    def endElement(self, name):
        if name == 'ArticleTitle':
            self.on = False

    def characters(self, content):
        if self.on:
            print content

if __name__ == '__main__':
    xml.sax.parse(open(sys.argv[1]), TitleAbstractContentHandler())
```