

1. Lähteet	2
2. Linkkejä	2
3. Tuntitehtävä	3
4. 00 Esipuhe	3
5. 01 Mikä ihmeen XML?	3
6. 02 Historiaa	5
7. 03 Välineet	6
8. 04 Hello World!	6
9. 05 Skandit mukaan	7
10. 06 Rakenneosat	8
11. 07 Eihän se näytä miltään	10
12. 08 CSS	10
13. 09 Ohjelmia css:n tekoon	12
14. 10 Tulostusongelmia?	13
15. 11 Tyhjät elementit	14
16. 12 Harjoittelua	15
17. 13 Linkit	15
18. 14 CDATA-osuudet	16
19. 15 DTD	16
20. 16 XML:ää ohjelmilla	17
21. 17 XSL	18
22. 18 Islandit	20
23. 19 XML-pohjaisia uutisia	20
24. 20 Tehtävä	21
25. 21 Arviointi	21

1.Lähteet

Lähteet

Boumphrey, F., Greer, C., Raggett, D., Raggett, J., Schitzenbaumer, S. & Wugofski, T. (2001). *XHTML*. Helsinki: Edita.

Eriksson, M. (1999). *XMLGuiden*. [Online: <http://www.acc.umu.se/~alpha/xml/>]. Luettu 9.1.2004.

IT Plus (2001). Oikea työkalu XML. *IT Plus* (3), 90.

Järvinen, J. (2004). Rss - XML-pohjaista uutistarjontaa. *Tietokone* (1), 76.

Keränen, V. (2002). Tee osoitekirja xml-kielillä. *Mikrobitti* (8), 68-71.

Keränen, V. (2003). Web-tyylit hallintaan. *Tietokone* (8), 75-75.

Korplea, J. (2003). *CSS-tyylit*. Jyväskylä: Docendo.

Körnerfors, R. (2003). *XML intro*. [Online <http://www.masda.vxu.se/multimedia/web/XMLintro/ex/land2/index.htm>]. Viitattu 1.2.2004.

North, S. & Herman, S. (2000). *XML. Parhaat ratkaisut ja olennaiset taidot*. Helsinki: Edita.

Nykänen, O. (2001). *XML*. Docendo: Jyväskylä.

Tuikka, T. & Kanala, S. (2001). *XML ohjelmoinnin perusteet*. Helsinki: Edita.

XML Introduction - What is XML. (2004). [Online: http://www.w3schools.com/xml/xml_what.asp]. Luettu 9.1.2004.

2.Linkkejä

Linkkejä

Sanakirja / sanastot

[Englanti - suomi -sanasto, v. 0.7.](#) Henri Ruinin XML-sanasto

XML

[Extensible Markup Language \(XML\) 1.0 \(Second Edition\)](#) : W3C:n suosituksia. Täyttä asiaa.

[Johdatus XML-tekniikkaan](#) : Henri Ruinin sivusto, jossa paljon faktoja. Ruini kirjoittaa esittelyssä: "Tämä on opas XML:n perustekniikkaan eli itse XML-määrittelyyn."

[Lyhyesti XML:stä](#) . Jukka Korpelan sivut.

[Lyhyt XML](#) : Lyhyesti XML:stä.

[Mikko Tarmian](#) kokoamia esimerkkejä

[Mikä on XML](#) : Jouni Heikniemiemen sivut, joissa on myös tyylä.

Rovasalo A. XML. Internet-julkaisussa: Rovasalo A, Tiedelehden toimittajan työkalupakki. URL: <http://www.helsinki.fi/people/Aki.Rovasalo/Tiedelehti/105.html> (3.5.2003)

[XML-opas](#) : Koulutus ja konsultointipalvelu KK .

[XML-peruskurssi](#) : Ossi Nykäsen materiaalia. Nykänen on mm. kirjoittanut kirjan aiheesta XML.

[XML 10 kohdan tiivistelmä](#) : Tästä on hyvä aloittaa. Ossi Nykäsen käännös alkuperäisdokumentista "Alkuperäisdokumentti XML in 10 points:".

[XML Euphoria in Perspective](#) : täyttä asiaa.

[XML Tutorial](#) : Microsoft opetta.

CSS

[CSS - Cascading Style Sheets](#)

[CSS-opas](#)

[CSS-opas](#)

[CSS1:n ominaisuudet](#)

[CSS - Tyylimäärittely](#)

[HTML/CSS koodit](#)

[Jojoxx's CSS guide](#)

[Värikkyttä: tyylisäännöt \(style sheets, CSS\)](#)

[Weppipakin CSS](#)

[Www-ohjelmointi: css-tyylit](#). Mikrobitti

RSS eli XML-pohjaisia uutisia

[Yle](#): Yleisradion Rss-sivu

HTML

[Esteettömien verkkosivujen tekemisen opas](#)

"Opas käsittelee sitä, miten verkkosivut saadaan mahdollisimman hyvin kaikkien ihmisten käytettäviksi, myös esimerkiksi vanhusten, lasten, vammaisten ja liikkeellä olevien."

[HTML-opas](#)

Jukka Packalénin ikivanha opas netissä. Moni asia on kuitenkin edelleenkin kuin vuonna 1997.

[HTML-opas](#)

Mika Vesterholmin tyylikkääät sivut.

[HTML-perusteet](#)

Sari Jaarasen tekemä nettikurssi.

[HTML tutorial](#)

Loistavat sivut html:n opiskeluun. Sivuilla on mm. html-editori, jonka avulla näet heti kuinka kirjoittamasi html-koodi toimii. Esimerkkejä opiskeluun on yhteensä 100 ja kaikki asiat käydään taatusti läpi.

[Makupalat: webjulkaiseminen](#)

Paljon hyödyllisiä linkkejä niin html:stä kuin muustakin webjulkaisemiseen liittyen.

[Näkövammaistahojen ohjeet verkkosivujen tarkistamiseen](#)

"Näkövammaiset on yksi ryhmä, joka voi selvästi hyötyä verkkopalveluiden lisääntymisestä. Verkkopalveluiden lisääntyvä käyttö edellyttää kuitenkin jatkossa suuremman huomion kiinnittämistä yleiseen käytettävyyteen, esteettömyyteen ja saavutettavuuteen."

[Publicera information på WWW \(HTML-kurs\)](#)

Ruotsin "opetushallituksen" sivut, jossa HTML:ää selostetaan ihan alkeista saakka.

[Resurser - HTML](#)

Ruotsiksi monenlaisia vinkkejä.

[W3C HTML Home Page](#)

HTML-kielen kotisivut verkossa.

Unicode

[Unicode merkitö](#): "Unicode on tietokoneissa käytettävä merkitö, johon on koodattu melkein kaikki maailman menneet ja nykyiset kirjaimistot, lukujärjestelmät ja keskeisimmät symbolit."

3. Tuntitehtävä

Tee sovellus, joka listaa seuraavat levyt nimen, esittäjän ja genren mukaan

The Album ABBA disco

Suuria kuvioita Anssi Kela iskelmä

Nummela Anssi Kela iskelmä

The very best of Bob Marley & The Wailers Bob Marley & The Wailers reggae

Yöperhonen Dreamville pop

Matkustaja Egotrippi pop

Reteesti vaan Irwin iskelmä

Taiteilijaelämää Ismo Alanko pop

Pelle show Kari Peitsamo iskelmä

Vapen & ammunition Kent pop

Lauluja rakastamisen vaikeudesta Leevi and the Leavings iskelmä

Torstai...40 seuraavaa hittiä Leevi and the Leavings iskelmä

Musiikkiluokka Leevi and the Leavings iskelmä

Kerran elämässä Leevi and the Leavings iskelmä

Hopeahääpäivä Leevi and the Leavings iskelmä

Tee oma rss-feed

4.00 Esipuhe

Esipuhe

Alun perin tein tämän materiaalin vain omaksi iloksi. Lähinnä tein muistiinpanoja, jotta muistaisin paremmin lukemani. Lisäksi kokeilin tietysti erilaisia esimerkkejä, jotka sitten tallensin. Vähitellen viehätyin XML:stä, koska se vaikutti helpolta. Lisäksi omat kotisivuni ovat XML:n ansiosta saaneet aivan uutta puhtia.

Kiinnostustani on viime aikoina kasvattanut ymmärrys siitä, että XML on tulevaisuuden kieli. Tosin samalla olen huomannut, että XML tarvitsee kaverikseen mm. HTML:ää ja CSS:ää. Vanhat opitut taidot eivät siis XML:ää käytettäessä pääse unohtumaan. Lisäksi XML on laajennettavissa, joten aina riittää lisää opittavaa. Tätä tehdessäni törmäsin esim. Rss:ään, joka on yksi XML-sovellus. Laitoin heti Rss dokumentin omalle sivulle ja hankin tietysti RssReaderin.

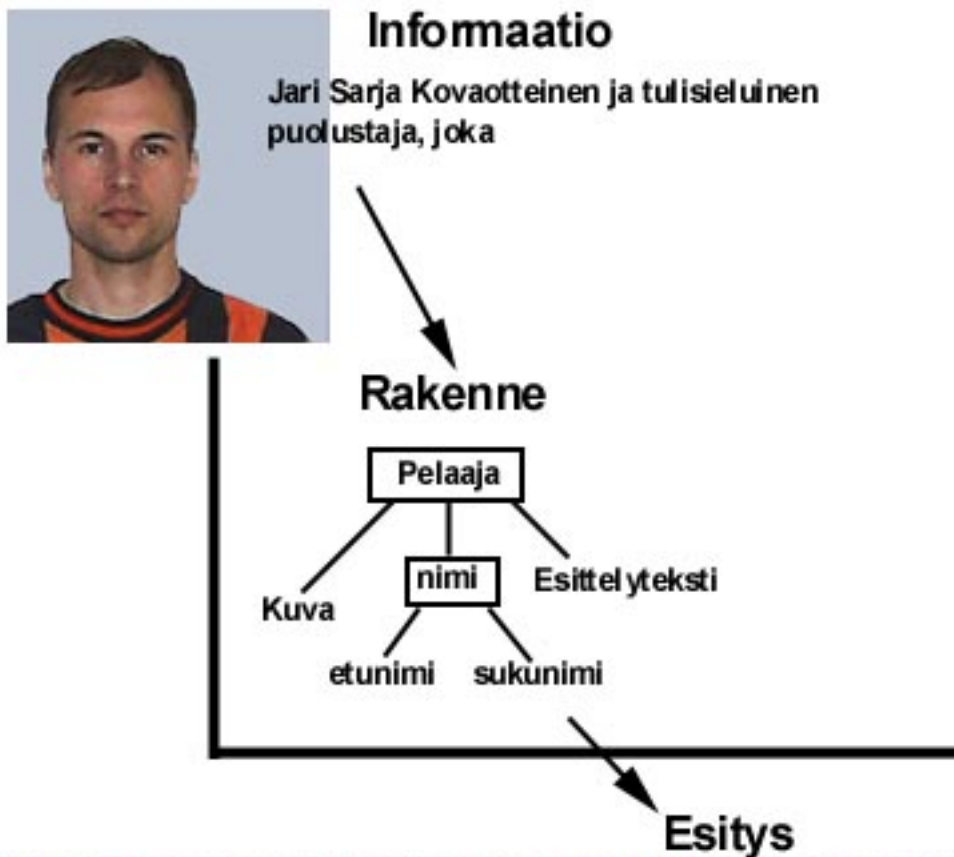
5.01 Mikä ihmeen XML?

1. Mikä ihmeen XML?

XML on lyhenne nimestä eXtensible Markup Language eli se on laajennettava merkintäkieli. Se on W3C:n (World Wide Web Consortium) suositus rakenteisten dokumenttien merkkäamiselle. Siitä odotetaan ratkaisua ongelmiin, jotka liittyvät elektroniseen julkaisemiseen ja tiedonsiirtoon. Innokkaimpien mielestä tulevaisuudessa on yhtä vaikea muistaa miten asiat tehtiin ennen XML:ää, kuten tällä hetkellä on muistaa miten asiat tehtiin ennen www:tä. (Nykänen 2001, 1; Tuikka & Kanala 2001, 1.) Itse asiassa XML on kieli, jolla voidaan määrittellä uusia kieliä. Sen avulla on tehty kieliä mm. molekyylien ja matemaattisten yhtälöiden määrittelyyn. (Keränen 2002, 68.)

XML on webin tulevaisuuden kieli

XML eli eXtensible Markup Language ratkaisee webissä tiedostoyhteensopivuusongelmat. XML on vähän kuten HTML, mutta ei sinne päinkään. HTML sisältää sekä datan että muotoilutiedot, mutta XML sisältää ainoastaan datan. XML:ssä visuaalinen taitto tallennetaan toiseen paikkaan. (IT Plus 2001, 90.) HTML siis ainoastaan esittää informaatiota eli se on tapa muokata informaation ulkoasua. Käyttämällä XML:ää voidaan informaatiolle luoda rakenne ja lisäksi voidaan päättää kuinka informaatio esitetään.



HTML-tagit on määritelty standardissa esim. tarkoittaa aina lihavoitua tekstiä. Kyseinen tägi ei tee muuta vaikka kuinka haluaisit sen tekevän. Et voi myöskään määrittellä omia tägejä tai määrittellä uudelleen vanhoja tägejä. XML puolestaan sallii sinun määrittellä omia tägejä tai tarkemmin sanottuna elementtejä. (IT Plus 2001, 90.)

Jos haluat XML:n sisältämän datan muotoiltuun muotoon on sinun otettava avuksi XSL-dokumentit (Extensible Stylesheet Language). XSL on enemmän kuin pelkkä tapa määrittellä tyylisivuja XML:lle. XSL osaa käsitellä dataa muutenkin kuin pelkän ulkoasun suhteen. Se osaa mm. aakkostaa, valikoida tietyt tiedot jne. Se on siksi paljon enemmän kuin pelkkä html tai css. (IT Plus 2001, 90.)

Miksi XML pystyy siihen mihin html ei pysty?

Ensinnäkin HTML:ssä ei ole syntaksin tarkistusta tai ainakaan sitä ei käytetä. Lisäksi selaimet hyväksyvät mitä tahansa html-koodia. Toiseksi HTML:stä puuttuu rakenteisuus. Kolmas html:n heikkous on, ettei siitä voi tunnistaa sisältöjä. Neljäs html:n heikkous on, ettei se sovellu datan vaihtoon. Viidenneksi on todettava, ettei html ole kansainvälistä. XML perustuu Unicodeen ja sen ansiosta XML selviytyy länsimaisten merkistöjen lisäksi myös aasialaisista kielistä. Kuudes huono puoli html:ssä on, ettei se ole oliopohjainen kuten monet nykyaikaiset ohjelmointikielät. Seitsemänneksi on huomattava, että html:stä puuttuu vankka linkitysmekanismi. Tämä ilmenee mm. siten, että kun yhden html:n tiedoston paikkaa vaihdetaan, joudutaan korjaamaan linkkejä moneen dokumenttiin. Kahdeksanneksi on todettava, ettei html:ää voi uusiokäyttää eli siis kierrättää. Yhdeksäs html:n heikkous suhteessa xml:ään on, ettei html ole laajennettavissa, sillä sitä ei koskaan edes suunniteltu sellaiseksi. (North & Hermans 2000, 9-11.)

Miten XML eroaa HTML:stä?

Suurin ero XML:n ja HTML:n välillä on siinä, että XML keskittyy sisältöön ja HTML keskittyy siihen miltä sisältö näyttää. XML ei tee muuta kuin strukturoi, säilyttää, kuvailee ja lähettää tietoa. XML:n erityispiirre on, että sinun pitää keksiä tagit itse. XML:ssä ei siis ole valmiita koodeja kuten HTML:ssä, mainittakoon esimerkkeinä tagit ja <p>. XML ei korvaa html:ää vaan tulevaisuudessakin niitä käytetään rinnakkain. XML kuvailee sisältöä, HTML puolestaan muotoilee ja näyttää sisältöjä. (XML Introduction - What is XML 2004.)

Tehtävä 1:

Tutustu XML:ään lukemalla dokumentti

"[XML 10 kohdan tiivistelmä](#)"

6.02 Historiaa

2. Historiaa ja taustaa

XML:n edeltäjiä olivat SGML ja HTML, jotka kumpikin ovat menestyneitä standardeja, kun kyseessä on dokumentin muotoilu. Sekä SGML:llä että HTML:llä on kuitenkin puutteita, sen vuoksi on kehitetty XML, joka siis yhdistää SGML:n ja HTML:n hyvät puolet. (Eriksson 1999.)

SGML

Standard Generalized Markup Language (SGML, ISO 8879) on kansainvälinen standardi sille kuinka dokumentteja merkataan. SGML on ollut olemassa jo vuodesta 1986, sekä HTML että XML ovat saaneet vahvasti vaikutteita SGML:stä. SGML on suosittu mm. puolustusvoimissa ja muissa isoissa teollisuuden aloilla, joissa käsitellään suuria määriä tietoja. SGML:llä on kuitenkin kaksi heikkoutta, nimittäin monimutkaisuus ja hinta. (Eriksson 1999.)

HTML

Päästäkseen eroon SGML:n monimutkaisuudesta ja kalliista työkaluista kehitti Tim Berners-Lee vuonna 1990 HTML -kielen. HTML:stä (HyperText Markup Language) tuli heti hitti. Oikeastaan kaikki web-sivut on tehty käyttäen html-kieltä, siksi html:n hallitsemista pidetään yleisesti niin tärkeänä. Kyseessä ei kuitenkaan ole mikään varsinainen ohjelmointikieli vaan lähinnä HTML on sivunkuvauskieli. Kyse on siis järjestelmästä, jonka avulla kuvataan dokumenttien sisältöä ja ulkoasua. HTML on myös paljon helpompaa oppia kuin oikea ohjelmointikieli. Lisäksi web-sivuja voidaan myös tehdä editorin avulla, jolloin web-sivujen tekeminen on lähes yhtä helppoa kuin tekstinkäsittely. Editoreita on kahta eri tyyppiä, toiset perustuvat koodin kirjoittamisen helpottamiseen, kun taas toiset ovat wysiwyg tyyppisiä.

HTML:n heikkous on sen rajoittuminen tageihin kuten <BODY>, <P> ja <TITLE>. HTML-kielessä ei siis voi luoda tagia <otsikko>, vaikka sattuisit sellaista tarvitsemaan. Sellaiset tagit kuten <I> ja ovat yksi syy siihen, että html:stä on muotoutunut kuvaileva kieli. Se määrittelee siis ulkonäköä eikä rakennetta. Tagi <I> ei kerro mitään sisällöstä, sen sijaan se kertoo, että sisältö kursivoidaan (i=italicize). Tagien vähäisyys sekä rakenteen ja muotoilun sekoittuminen aiheuttavat ongelmia. Lisäksi HTML-kieli on täynnä lyhenteitä, jotka eivät kerro mitään (p, br, td, ol, li), siksi HTML-koodia on niin vaikea lukea ja tulkita. Netscape ja Microsoft, jotka valmistsivat web-selaimia ovat yrittäneet poistaa HTML:n puutteita luomalla omia laajennuksiaan. Seurauksena on kuitenkin ollut yhteensopimattomuus eri selainten välillä. Websivulla oleva ilmoitus "Tämä sivu näkyy parhaiten Internet Explorerilla" on yhteensopimattomuudesta johtuva ikävä seuraus. (Eriksson 1999.)

XML

HTML on kehitetty SGML:stä eli se on tehty noudattaen SGML:n sääntöjä. XML on sitä vastoin osa SGML:ää. Kyseessä on suuri ero sillä XML:n avulla voidaan määrittellä uusia sovelluksia, sellaista ei voi

tehdä HTML:n avulla. XML:llä tehtyjä sovelluksia ovat mm. Mathematical Markup Language (MathML) ja Channel Description Format (CDF). Tässä onkin XML:n suurin vahvuus eli se on laajennettavissa. Eri organisaatiot voivat kehittää omia sovelluksiaan, omiin tarpeisiinsa. (Eriksson 1999.)

XML:n vahvuudet verrattuna HTML:ään

Laajennettavuus: Mahdollisuus luoda omia elementtejä ja attribuutteja.

Rakenne: XML mahdollistaa tiedon loogisen strukturoinnin.

Validointi: XML:n dokumentin oikeellisuus on helppo tarkistaa. (Eriksson 1999.)

Tehtävä 2

Suorita koneellasi haku *.xml

Tutki löydökset eli mitä xml-dokumentteja löytyi koneeltasi?

7.03 Välineet

3. Välineet

XML-ohjelmointi onnistuu ihan kotikonstein. Tarvitset ainoastaan selainohjelman, joka ymmärtää XML:ää. Uusimpiin selaimiin sisältyy XML-jäsennin, joka hoitaa XML-dokumenttien käsittelyn. XML:ää ymmärtävä selain on esimerkiksi Internet Explorer (versio 5 ja uudemmat) tai Netscape Navigator (versiosta 6 alkaen). Parhaiten XML toimii tällä hetkellä Internet Explorerin versiolla 6.

XML-koodin kirjoittaminen onnistuu pelkällä muistolla. Käytännössä XML-dokumenttien hallinta on helpompaa kunnollisella XML-muokkaimella. Alkuun pääsee kuitenkin hyvin pelkällä muistiolla ja usein nopein tapa kirjoittaa koodia on juuri tehdä se muistiolla. Myöhemmässä vaiheessa kurssia käsittelemme XML-muokkaimia lähemmin.

8.04 Hello World!

4. Hello world!

Yleinen tapa aloittaa ohjelmointi on alkaa se esimerkillä "Hello world!". Siksi tässäkin noudatetaan samaa kaavaa. Alla on yksinkertainen XML-dokumentti, joka siis tulostaa näytölle tekstin "Hello World!"

```
<?xml version="1.0"?>
<esimerkki tyyppi="pakollinen">
<tervehdys>Hello World!</tervehdys>
</esimerkki>
```

XML-dokumentin alussa on aina prosessointiohje, joka kertoo xml-kielen version ja dokumentin käyttämän merkistön. Ohjeen on oltava xml-dokumentin ensimmäisellä rivillä ja se kirjoitetaan merkintöjen <? ja ?> väliin. XML on kirjoitettu pienin kirjaimin, eikä kysymysmerkkien kummallakaan puolella ole välilyöntiä. Ainoa käytettävissä oleva xml-kielen versio on toistaiseksi 1.0.

Elementti

```
<tervehdys>Hello World!</tervehdys>
```

Kussakin XML-dokumentissa saa olla vain yksi juurielementti. Tässä esimerkissä kyseinen elementti on <esimerkki>. Kaikkien muiden elementtien täytyy olla juurielementin sisällä.

Attribuutti

```
<esimerkki tyyppi="pakollinen">
```

Elementtitunnisteet voivat sisältää yhden tai useamman attribuutin, jotka antavat lisätietoja elementeistä. Tässä elementin <esimerkki> attribuuttina on pakollinen.

Tehtävä 3.

Kokeile yllä olevaa esimerkkiä omalla koneellasi.

XML ei salli kirjoitusvirheitä eli loppu- ja alkutunnisteen on oltava samanlaiset. Lisäksi selain pysähtyy siihen kohtaan, jossa virhe ilmenee. Alla olevassa esimerkissä on kirjoitettu väärin lopputunniste < esimerkki>. Siitä seuraa tällainen virheilmoitus

Lopputunniste esimerkki ei vastaa alkutunnistetta esimerkki.
Virhe resurssia <http://sarja.internetix.fi/fi/sisalto/materiaalit/gfx/xml/hello.xml>
käsiteltäessä. Rivi 4, paikka 3
</esimerkki>
--^

Sen jälkeen ei auta muu kuin korjata virhe ja avata "uusi" selain. Ainakaan minun koneessa ei pelkkä reload riittänyt.

Mikäli kaikki meni oikein tulisi näkymän olla tällainen kuten tässä alla.

```
<?xml version="1.0" ?>  
- <esimerkki tyyppi="pakollinen">  
  <tervehdys>Hello World!</tervehdys>  
</esimerkki>
```

Tässä vaiheessa näkymä ei ole kummoinen, koska xml-dokumenttiin ei vielä ole liitetty mitään tyyliä (vrt. kuva luvussa 1). Lisäksi on huomioitava, että vanhemmilla selaimilla (esim. vanhemmat kuin Explorer 5) et välttämättä näe mitään.

9.05 Skandit mukaan

5. Skandit mukaan

Skandinaaviset merkit kuten å, ä, ö, Å, Ä ja Ö eivät toimi suoraan ilman muutoksia. Seuraava esimerkki ei siis toimi:

```
<?xml version="1.0"?>  
<esimerkki tyyppi="pakollinen">  
<tervehdys>Töttöröö maailma!</tervehdys>  
</esimerkki>
```

Tehtävä 4.

Kokeile yllä olevaa esimerkkiä.

Yllä olevasta esimerkistä seuraa virheilmoitus, joka on tällainen kuin tässä alla.

XML-sivua ei voi näyttää

XML-syötettä ei voi näyttää, jos tyyllisivu on XSL. Korjaa virhe ja valitse [Päivitä](#) tai yritä myöhemmin uudelleen.

Teksti sisältää merkin, joka ei kelpaa. Virhe resurssia
<http://sarja.internetix.fi/fi/sisalto/materiaalit/gfx/xml/teht2>
käsiteltäessä. Rivi 3, paikka 13

```
<tervehdys>T
```

Kirjoita ensimmäinen rivi uudestaan seuraavalla tavalla:

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

Tässä on siis määritelty merkistö, joka tässä tapauksessa on siis ISO-8859-1, joka on länsieurooppalaisten kielten käyttöön jo 80-luvulla sovittu 8-bitin standardi Iso-Latin-1. Tällä merkistöllä voidaan ongelmitta käsitellä lähes kaikkia länsieurooppalaisissa kielissä esiintyviä erikoiskirjaimia.

Muutoksen jälkeen sivun tulisi toimia [näin](#).

Tehtävä 5.

Tee muutos ja kokeile toimivatko skandit.

10.06 Rakenneosat

6. XML:n rakenneosat

Seuraavaksi tarkastellaan XML:n rakennetta esimerkin avulla:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<!DOCTYPE dokumentti [
<!ENTITY copy "Tämän on tehnyt Jari Sarja">
]>
<dokumentti>
<henkilo>
<nimi>Jari Sarja</nimi>
<sposti>jari.sarja@sarja.fi</sposti>
<pituus yksikko="cm">179</pituus>
<suosikkitagi>&lt;P&gt;</suosikkitagi>
</henkilo>
<henkilo>
<nimi>Otto von Töttenberg</nimi>
<sposti>otto@hotentottimaili.fi</sposti>
<pituus yksikko="cm">170</pituus>
<suosikkitagi>&lt;B&gt;</suosikkitagi>
</henkilo>
<loppuun>&copy;</loppuun>
</dokumentti>
<!-- Tämä esimerkki on muunnos M.Erikssonin esimerkistä -->
```

Esittelyosa

XML-dokumentin ensimmäisellä rivillä täytyy määritellä XML-versio, jota käytetään. Tällä hetkellä on olemassa ainoastaan versio 1.0, mutta tulevaisuudessa W3C saattaa kehittää uusia versioita.

```
<?xml version="1.0"?>
```

Tämä yllä oleva määrittely on yksinkertaisin ja kertoo siis käytettävän version. `<? ja ?>` määrittelevät, että kaikki se mitä on siinä välillä on n.k. processing instructions (PIs). Kyse on on ohjeista sille ohjelmalle, joka käsittelee dokumenttia. Versionumeron on oltava lainausmerkkien sisällä , `version="1.0"`. Tällä hetkellä ainoa käytettävissä oleva XML-kielen versio on 1.0, jonka oletusmerkistö on Unicode.

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

Yllä olevaan määrittelyyn on lisätty kaksi attribuuttia, josta `encoding` kertoo XML-ohjelmalle mitä merkistöä (character set) dokumentissa käytetään. Mikäli kirjoitat suomeksi tai jollain muulla länsieurooppalaisella kielellä, on suositeltavaa käyttää merkistöä Latin-1 (ISO-8859-1). `Standalone` kertoo sen, että onko dokumentti valmis sellaisenaan vai tuodaanko siihen vielä tietoja toisesta tiedostosta.

Kommentit

XML-dokumenttiin voi kirjoittaa kommentteja samalla tavalla kuin HTML-dokumenttiin, käytössä ovat tagit `<!-- ja -->`. Kommenttien tarkoitus on ainoastaan, että niihin sisällytetään tietoa. XML:ssä ei siis voi tehdä kuten HTML:ssä, että kyseisten koodien väliin sijoitetaan JavaScript tai VBScript koodia.

```
<!-- Tämä esimerkki on muunnos M.Erikssonin esimerkistä -->
```


Kussakin XML-dokumentissa saa olla vain yksi juurielementti ja kaikkien muiden elementtien on oltava täysin sen sisällä.

```
<dokumentti>
<!-- Siinä juurielementti -->
</dokumentti>
```

Tässä esimerkissä juurielementin nimi oli siis "dokumentti".

Elementit

XML:ssä elementti koostuu normaalista kolmesta asiasta: alkutunnisteesta, sisällöstä ja lopputunnisteesta.

```
<henkilo>
<nimi>Jari Sarja</nimi>
<sposti>jari.sarja@sarja.fi</sposti>
</henkilo>
```

Hyvin muodostetussa XML-dokumentissa elementit muodostavat hierarkian, jolloin jokaisella elementillä on vain yksi emoelementti. Tässä esimerkissä elementit nimi ja sposti ovat henkilö-elementin lapsia (children). Henkilö elementti on vanhempi (parent).

XML tekee eron suurten ja pienten kirjaimien välillä. Et siis voi kirjoittaa näin:

```
<Nimi>Jari Sarja</nimi>
```

Lisäksi on huomioitava, että tagit eivät saa mennä päällekkäin.

Ei siis näin:

```
<nimi>Jari<osoite></nimi>Putkinotkontie 4</osoite>
```

vaan näin:

```
<nimi>Jari</nimi><osoite>Putkinotkontie 4</osoite>
```

Entiteetit

Entiteetit sisältävät yleensä tekstiä, mutta sisältö voi myös olla ääntä tai kuvaa. Entiteettien tarkoitus on helpottaa kirjoittamista. Tässä esimerkissä olin määritellyt entiteetin: copy, jonka liitin sitten loppuun komennolla ©.

```
<!DOCTYPE dokumentti [
<!ENTITY copy "Tämän on tehnyt Jari Sarja">
]>
<loppuun>&copy;</loppuun>
```

XML:ssä on joukko ennalta määriteltyjä entiteettejä:

& = &

' = '

> = >

> = <

" = "

Nämä ovat merkkejä, jotka tulisi korvata. Muutoin se sekoittuvat merkkäusmerkkeihin.

Ei siis näin paina <enter>

vaan paina >enter<

```
<suosikitagi>&lt;P&gt;</suosikitagi>
```

Attribuutit

Attribuutin tehtävänä on kertoa lisää tietoa elementistä. Attribuutit luetellaan aina elementin alkutunnisteessa. Attribuutit koostuvat nimestä ja sen arvosta. Arvo annetaan aina lainausmerkkien sisällä. Yhdellä elementillä voi olla useita arvoja.

```
<pituus yksikko="cm">179</pituus>
```

Tehtävä 6.

Kokeile sivun alussa olevaa esimerkkiä omalla koneellasi. Perehdy merkkauksiin ja niiden merkityksiin.

11.07 Eihän se näytä miltään

7. Eihän se näytä miltään

XML-dokumentti määrittelee sisällön rakenteen. Se ei ota kantaa siihen miltä sivu näyttää, kun sitä katsotaan esim. webselaimen kautta. Jos siis haluaa selaimen kautta nähdä muutakin kun rakennepuun, voi ongelman helpoiten ratkaista käyttämällä css-tyyliä (Cascading Style Sheets) eli tyyliohjetta. Tee siis esim. tällainen tyyliohje:

```
henkilo {
display: block;
height: auto;
width: 300px;
border-color: #000000;
background-color: #CCCCCC;
border-style: solid;
border-width: 1px;
left:4px;
margin:4px;
padding:4px;
}
nimi {
font-family: Arial, Helvetica, sans-serif;
font-size: 12pt;
font-weight: bold;
color: #990000;
}
sposti, pituus, suosikkitagi {
display: block;
font-family: Arial, Helvetica, sans-serif;
font-size: 10pt;
color: #000000;
}
```

Tee tyylisivuun viittaus esim. näin:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<?xml-stylesheet href="tyylisivu.css" type="text/css"?>
```

Oikea paikka on siis XML määrittelyn jälkeen. HREF ilmaisee missä tyyli-dokumentti sijaitsee. Tässä esimerkissä css-tiedosto "tyylisivu.css" sijaitsee siis samassa kansiossa kuin varsinainen xml-dokumentti. Lopputulos on sitten kutakuinkin [tällainen](#).

Tehtävä 7.

Kirjoita tyylisivu ja lisää se edellä olevaan esimerkkiin.

12.08 CSS

8. CSS

"CSS:n merkitys kasvaa kahdesta syystä. Se tarjoaa web-sivujen tekijöille joustavan mahdollisuuden vaikuttaa sivujen ulkoasuun. Itse asiassa CSS:n jonkinasteinen oppiminen on sivuntekijälle lähes

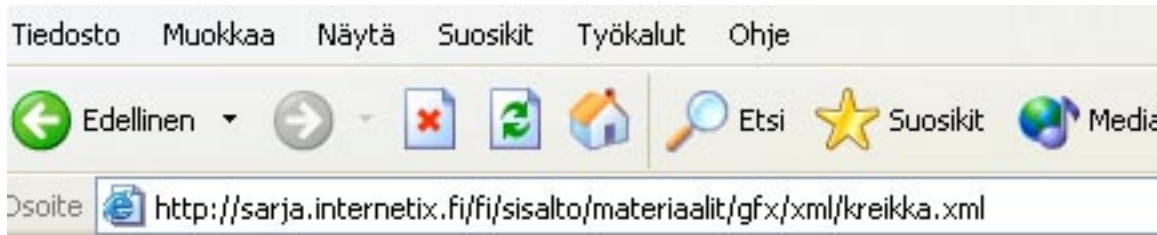
välttämätöntä, ja mitä vaikuttavampi ulkoasu halutaan, sitä tarkemmin pitää tuntea CSS:n mahdollisuudet ja niiden käyttö. Toiseksi CSS tarjoaa yksinkertaisen tavan ilmoittaa XML-tiedoston esitysmuoto, ja tämän merkitys kasvaa sitä mukaa kuin XML käytännössä yleistyy." (Korpela 2003, IX)
Käytetään esimerkkinä edellistä XML dokumenttia.

Taitto

Laitetaan koko tekstin tausta harmaaksi tekemällä määriykset koskien juurielementtiä "dokumentti".
Laitetaan vielä tekstin väriksi musta.

```
dokumentti { background: gray;  
color: black; }
```

Lopputulokset tässä alla.



```
Jari Sarja jari.sarja@sarja.fi 179 Otto von Töttenberg otto@hotentottimaili.fi 170
```

Kuten huomaat, teksti on pötkössä. Näin siksi, että XML-dokumenteissa on alkuarvona inline. Siksi lohkoesitys on muutettava muotoon block, jolloin jokainen elementti on omalla rivillään.

```
dokumentti { background: gray;  
color: black;  
}  
nimi, posti, pituus { display: block;  
}
```

Näyttää siis tältä:



```
Jari Sarja  
jari.sarja@sarja.fi  
179  
Otto von Töttenberg  
otto@hotentottimaili.fi  
170  
Tämän on tehnyt Jari Sarja
```

Elementille dokumentti ei vielä ole määritelty kokoa eikä kehystä. Laitetaan siis kehykset mukaan ja tehdään vähän varjomaista vaikutelmaa.

```
dokumentti { background: gray;  
color: black;  
width: 300px;  
border-style: solid outset outset solid;  
padding: 0em 0em 0em 0em;
```

```
border-width: thin 0.2em 0.2em thin;
}
nimi, sposti, pituus {
display: block; }
```

Lopputulos tässä alla.



```
Jari Sarja
jari.sarja@sarja.fi
179
Otto von Töttenberg
otto@hotentottimaili.fi
170
Tämän on tehnyt Jari Sarja
```

Tekstin muotoilu

Kirjasinlaji voidaan määrittellä font-family ominaisuudella.

```
nimi {font-family: "Trebuchet MS", Arial, sans-serif; }
```

Fontin koko voidaan asettaa monella eri tapaa. Tässä pari esimerkkiä:

```
nimi { font-size: 110%; }
sposti {font-size: medium; }
pituus { font-size: 12pt; }
```

```
nimi { font-size: 1.2em; }
sposti {font-size: large; }
pituus { font-size: 12px; }
```

Fontin lihavuutta on css:ssä olemassa peräti yhdeksän kappaletta ja ne on numeroitu 100:sta 900:aan sadan välein. Tässä esimerkkiä.

```
nimi { font-weight: 300; }
```

Tehtävä 8.

Muokkaa css:ää, jotta lopputulos olisi kutakuinkin [tällainen](#).

13.09 Ohjelmia css:n tekoon

09 Ohjelmia tyyliohjeitten tekoon

Tyyliohjeita voidaan kirjoittaa käsin koodaamalla, mutta nykyisin on myös olemassa ohjelmia, joilla voidaan helposti kirjoittaa tyyliohjeita. Nykyisin on myös olemassa sivunkeko-ohjelmia, joilla voidaan tehdä tyyliohjeita. Parhaimmillaan tyyliohjeen laatiminen on sitä, että ohjelma näyttää lopputuloksen jo sivunkekoaiheessa. (Keränen 2003, 75.)

Dreamweaver MX

Tyyliohjeita luodaan ja muokataan CSS Styles -paneelissa. Aluksi on luotava uusi tyyliohje "New CSS Style" - toiminnolla ja määriteltävä kuinka tyyli liitetään sivulle. Tyyliohjeesta voidaan tehdä sivukohtainen tai erillinen tyyli tiedosto. HTML-elimet määritellään uudelleen "Refine HTML Tag -toiminnolla. Luokan voi muodostaa "New CSS Style -ikkunassa valitsemalla tyyli tyyppiä "Make Custom Style". (Keränen 2003, 75.)

TopStyle lite (ilmainen)

(<http://www.bradsoft.com/download/index.asp>)

TopStyle-ohjelma on helppokäyttöinen ja se osaa lisäksi kertoa eri selainten ymmärtämät tyyliominaisuudet. Käyttämällä kyseistä ohjelmaa, vältät pahimmat tyyliominaisuudet. HUOM! [Mikrobitin](#) sivuilta on saatavissa vielä myös vanhempi versio eli TopStyle 2.5.

Validaattori

CSS:n yhteydessä on tärkeää kirjoittaa koodia oikein, koska selainten muutenkin puutteellinen CSS-tuki lakkaa usein toimimasta, jos merkkaukset ei ole oikeita. (Korpela 2003, 46.)

[W3C CSS Validation Service](#)

14.10 Tulostusongelmia?

Tulostus

Yleinen ongelma taulukkosivujen kanssa on tulostus. Usein käy niin, että teksti leikkautuu ikävästi oikeasta reunasta. Lisäksi navigaatiot tulevat mukaan ja kaikki linkit on alleiviivattu. Tekstiä on siis erittäin hankala lukea.

Teksti saattaa paperilla olla esim. tämän näköistä kuin tässä alla, vaikka se ruudulla näkyisikin oikein.

artikkelit . mediat . työkalut . opiskelu . yhteisöt . et

Eksosomaattinen oppiminen

Fyysisten ja kielellisten välineitten avulla ihminen voi ylittää itsensä. Vuorovaikutuksemme kyseisten välineiden kanssa on keskeistä oppimisen kannalta. Yhdessä nämä erilaiset väline muodostavat osan kulttuurisista voimavaroistamme. Jo pikainen silmäys omaan asuntoomme osoittaa kuinka tärkeiksi erilaiset fyysiset välineet ovat tulleet erilaisten toimintojen ylläpidossa ja erilaisten ongelmien ratkaisuisissa. Elämme autojen, puhelinten, CD-soittimien, mikroaaltouunien, televisioiden, kahvinkeitinien ja tietokoneiden maailmassa. (Säljö 2001, 72.) Nämä kaikki em. välineet sijaitsevat oman kehomme ulkopuolella, ne ovat siis eksosomaattisia.

Ratkaisu

Ongelman voi ratkaista käyttämällä css-tyyliä (Casacading Style Sheets) eli tyyliohjetta. Ainakin Explorer 6 ja Netscape 7 tukevat tulostuksen erillistä ohjausta ccs:n avulla. Toisaalta esim. Netscape 4.7. ei ymmärrä ollenkaan css kakkosta. Tee siis esim. tällainen tyyliohje ja tallenna se juuritasolle tai johonkin muuhun kansioon.

```
@media print {  
.noprnt {display: none; }  
a { text-decoration: none;  
color: black;  
background: white; }
```

```

table { border : none;
border-bottom : none;
border-style : none;
text-align : left; }
TD { color : Black;
border : none;
background-color : white;
font-family : Times New Roman, serif;
font-size : 13pt;
text-align : left; }
}
@media screen {
a { text-decoration: none;
color: black;
background: white; }
table { border : none;
border-bottom : none;
border-style : none;
text-align : left; }
TD { color : Black;
border : none;
background-color : white;
font-family : Times New Roman, serif;
font-size : 13pt;
text-align : left; }
}

```

Tee tyylisivuun viittaus esim. näin:

```

<head>
<title>IxApplication test</title>
<LINK REL="StyleSheet" TYPE="text/css" HREF="/fi/testi.css"/>
</head>

```

Oikea paikka on siis heti titlen jälkeen. HREF ilmaisee missä tyylidokumentti sijaitsee. Tässä esimerkissä css-tiedosto "testi.css" sijaitsee siis ihan juuritasolla. Lopputulos on sitten kutakuinkin [tällainen](#). HUOM! sivun idea paljastuu vasta, kun sen tulostaa tai katselee sivua tulostuksen esikatselussa. Homman voi tehdä toisinkin, jolloin eri medioille tehdään erilliset tyylitiedostot. Esim. näin:

```

<link rel="stylesheet" type="text/css"
href="/css/testi1.css" media="print" />
<link rel="stylesheet" type="text/css"
href="/fi/testi2.css" media="screen" />

```

Tehtävä 10.

Kopioi sivu

<http://sarja.internetix.fi/fi/sisalto/verkkopeda/kuvaoppimateriaalissa/luku1>

Muokkaa sivua niin, ettei tulostettaessa näy vasemman puoleista navigaatiovalikkoa eikä yläpalkkia ja määrittele tulostuksen fontiksi Times New roman.

15.11 Tyhjät elementit

10. Tyhjät elementit

XML:ssä tyhjässä elemetissä käytetään kauttaviivaa (/) ennen viimeistä kulmasulkumerkkiä (>) Boumphrey, F., Greer, C., Raggett, D., Raggett, J., Scitzenbaumer, S. & Wugofski, T. (2001, 280).

```
Lopussa siis kauttaviiva /  
  
<BR/>
```

16.12 Harjoittelua

12. Harjoittelua

Tehtävä 12.

Tee alla olevasta tekstistä XML-dokumentti, jotta siitä tulisi [tällä inen](#).

Eksosomaattinen oppiminen

Fyysisten ja kielellisten välineitten avulla ihminen voi ylittää itsensä. Vuorovaikutuksemme kyseisten välineiden kanssa on keskeistä oppimisen kannalta. Yhdessä nämä erilaiset välineet muodostavat osan kulttuurisista voimavaroistamme. Jo pikainen silmäys omaan asuntoomme osoittaa kuinka tärkeäksi erilaiset fyysiset välineet ovat tulleet erilaisten toimintojen ylläpidossa ja erilaisten ongelmien ratkaisuisissa. Elämme autojen, puhelinten, CD-soittimien, mikroaaltouunien, televisioiden, kahvinkeitinien ja tietokoneiden maailmassa. (Säljö 2001, 72.) Nämä kaikki em. välineet sijaitsevat oman kehomme ulkopuolella, ne ovat siis eksosomaattisia.

Oppimisen suhteen onkin kysyttävä, että onko tekniikka ihmisen sisä- vai ulkopuolella? Toinen yhtä keskeinen kysymys on, että onko oppiminen aina samanlaista? Tekniikan kehittyessä meidän kommunikointimme on muuttunut, samalla tavalla myös oppiminen on muuttunut. Oppiminen eilen ja tänään eivät siis ole täysin samanlaisia tapahtumia. Onkin hyvä luoda katsaus taaksepäin, ja tarkastella sitä kuinka ihmiset ennen kommunikoivat ja oppivat. Yksi olennainen ero eilispäivän ja nykypäivän välillä on siinä, että aina ei ole ollut instituutiota nimeltä koulu. Kuitenkin myös ennen ihmiset kommunikoivat ja oppivat.

17.13 Linkit

13 Linkit

XLink -linkit

XLink eli Extensible Linking Language on jälleen yksi lisää laajaan XML perheeseen, jota W3C on kehittänyt. XLink koostuu kahdesta osasta: XLink (XML Linking Language) ja XPointer (XML Pointer Language). XLink mahdollistaa ulospäin suuntautuvat linkit ja linkitykset eri suuntiin, kun taas XPointer kuvaa sitä kuinka XML-dokumentin sisällä linkitetään. (Eriksson 1999.)

XLink

HTML:ssä linkki määritellään tagilla <A>. XML:ssä mikä tahansa tagi eli siis elementti voi olla linkki. Parseri tunnistaa linkin attribuutit xml:link, jolla voi olla arvona joko simple, extended, locator, group tai document, riippuen siitä millaisesta linkistä on kyse. Ihan samalla tavalla kuin HTML:ssä niin attribuutti href määrittelee sen minne linkitetään. Tässä alla pari esimerkkiä. (Eriksson 1999.)

```
<kuva xml:link="simple" href="jari.jpg"/>  
<artisti xml:link="simple" href="http://zappa.com">  
Frank Zappa</artisti>
```

Nämä esimerkit muistuttavat paljolti HTML-linkkejä. Kyseisen tyyppiset nk. Simple Links ovat yhdensuuntaisia linkkejä, ihan kuten html:ssä.

Selaimet tukevat XLinkkiä huonosti, joten siksi yllä oleva esimerkki ei toimi esim. Internet Explorerilla (versio 5 tai 6). Linkkiongelman voi kuitenkin ratkaista käyttämällä nimiavaruutta. Sen avulla on mahdollista liittää XML-dokumentin sisään myös HTML-koodia. (Tuikka & Kanala 2001, 36.)

```
<?xml version="1.0" ?>  
<?xml-stylesheet type="text/css" href="tyyli.css" ?>  
<linkkilista xmlns:html="http://www.w3.org/TR/REC-html40">  
<nimi>Jari Sarjan kotisivu</nimi>  
<linkki>
```



```
<html:a href="http://sarja.internetix.fi">
<osoite>http://sarja.internetix.fi</osoite>
</html:a>
</linkki>
<kuvaus>Jarín Nexuksella tehdyt huippusivut</kuvaus>
<nimi>Jukka Tikkasen kotisivu</nimi>
<linkki>
<html:a href="http://jukkatikkanen.internetix.fi">
<osoite>http://jukkatikkanen.internetix.fi</osoite>
</html:a>
</linkki>
<kuvaus>Jukan Nexuksella tehdyt huippusivut</kuvaus>
</linkkilista>
```

Tehtävä

Kirjoita esimerkki ja tee siihen tyylitiedosto, jotta lopputulos olisi kutakuinkin [tällainen](#).
Kuvan saa mukaan samanlaisella tempulla eli näin.

```
<html:img src="http://www.internetix.fi/jari/jari.jpg"/>
```

Linkeissä ja kuvissa töytyy sinne dokumentin alkuun kirjoittaa:
<linkkilista xmlns:html="http://www.w3.org/TR/REC-html40">

Tehtävä

Kirjoita esimerkki ja tee siihen tyylitiedosto, jotta lopputulos olisi kutakuinkin [tällainen](#).

18.14 CDATA-osuudet

14 CDATA-osuudet

Mikäli haluat kirjoittaa koodia XML-dokumenttiin, jota ei prosessoida XML:n tapaan, käytä silloin CDATA-osuutta (character data). Tällöin voi kirjoittaa vaikka javascript -koodia, jonka haluat esittää sellaisenaan. Parseri ei yritä tulkita millään tavalla CDATA-lohkon sisältöä.

```
alku <![CDATA[
loppu ]]>
```

Esim.

```
<?xml version="1.0"?>
<esimerkki tyyppi="pakollinen">
<tervehdys>Hello World!</tervehdys>
<![CDATA [ <a href="#" onclick=
"javascript:window.print()">Tulosta sivu!</a>
]]>
</esimerkki>
```

Tehtävä.

Kokeile yllä olevaa esimerkkiä.

19.15 DTD

DTD

DTD on lyhenne sanoista Document Type definition. Sen tehtävä on kuvailla XML-asiakirjaa eli sääntöjä, jotka määrittelevät tiedoston rakenteen. DTD ei ole pakollinen, mutta sen avulla esim. webiseläin voi tarkistaa saamansa XML-datan yhteensopivuuden ja oikeellisuuden.

DTD on ratkaisu ongelmiin, jotka liittyvät tiedon löytämiseen. Esimerkiksi jos etsimme webistä tietoa sellaisista maista, joiden väkimäärä on yli 200 miljoonaa, on lähes mahdotonta tietää kuinka sivut on rakennettu. Joku on antanut tagille nimen "ASUKKAITA", toinen on antanut nimeksi "ASUKASLUKU" ja kolmas käyttää tagia "VÄKIMÄÄRÄ". (Körnerfors 2003.)

Jotta etsittävä tieto löytyisi, pitää siis sopia rakenteesta ja tagien nimistä. Se on ratkaistavissa käyttämällä DTD:tä (Document Type Definition). DTD määrittelee tagien nimet ja kuinka niitä käytetään. Jos haluamme, että sivustomme on yhtenäinen kaikilta osin, siis myös tiedon käytettävyyden kannalta, käytämme siis DTD:tä. (Körnerfors 2003.)

DTD:n määrittelee siis tagien nimet ja niiden hierarkiset suhteet. Sen avulla voimme varmistua, että rakenne rakennetaan samalla tavalla, jonka seurauksena tiedon hakeminen ja vertailu helpottuvat. Sen jälkeen on enää yksi ongelma ratkaisematta eli se kuinka tieto kirjoitetaan. Jos olemme sopineet, että käytämme tagia "ASUKASLUKU" niin joku voi kirjoittaa 200 miljoonaa, toinen kirjoittaa 200 000 000 ja kolmas kirjoittaa 200000000. Tätä ongelmaa ei voi ratkaista pelkästään käyttämällä DTD:tä vaan avuksi on otettava XML-skeemat. (Körnerfors 2003.)

Esimerkki sisäisestä DTD:stä.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="tyyli.css" ?>
<!DOCTYPE puhluettelo
[
<!ELEMENT osoitekirja (henkilo)*>
<!ELEMENT henkilo (etunimi,sukunimi,pnumero?)>
<!ELEMENT etunimi (#PCDATA)>
<!ELEMENT sukunimi (#PCDATA)>
<!ELEMENT pnumero (#PCDATA)>
]
>
<puhluettelo>
<henkilo>
<etunimi>Jari</etunimi>
<sukunimi>Sarja</sukunimi>
<pnumero>015-464213</pnumero>
</henkilo>
<henkilo>
<etunimi>Jari</etunimi>
<sukunimi>Sarja</sukunimi>
<pnumero>015-464213</pnumero>
</henkilo>
</puhluettelo>
```

Ulkoisen DTD

```
<?xml version="1.0"?>
<!DOCTYPE puhluettelo SYSTEM "puhluettelo.dtd">
```

? elementti voi esiintyä 0 tai 1 kertaa
+ elementti voi esiintyä 1 tai monta kertaa
* elementti voi esiintyä 0 tai monta kertaa

Tehtävä

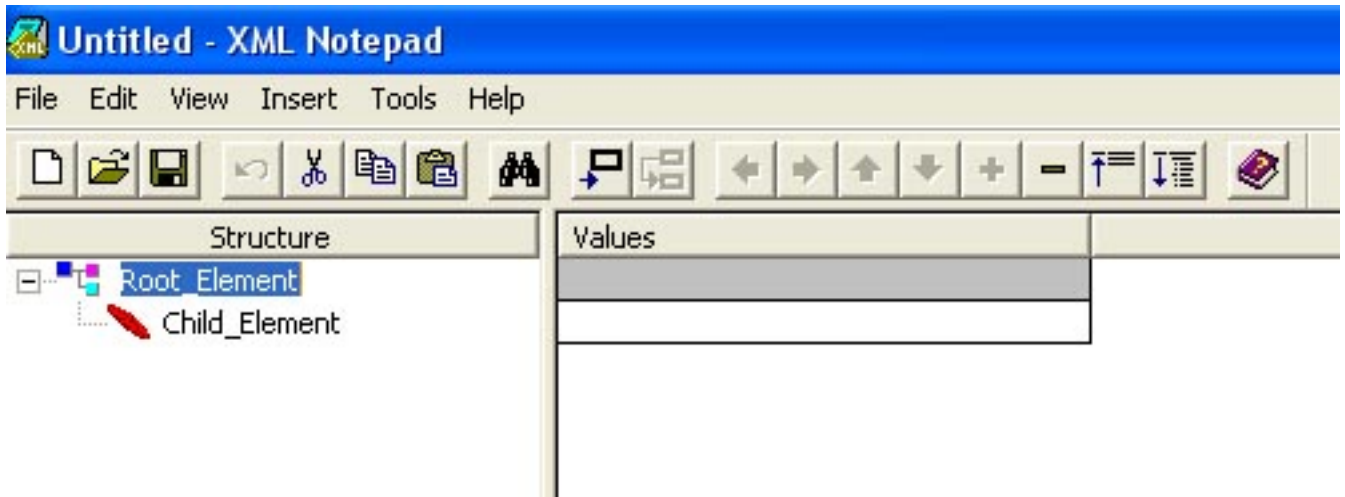
kokeile yllä olevaa esimerkkiä sisäisestä DTD:stä

16 XML:ää ohjelmilla

XML:ää voi kirjoittaa pelkällä muistolla, mutta nykyisin on myös olemassa ohjelmia, joilla voi kirjoittaa XML:ää.

Ohjelmia

"XML Notepad on Microsoftin ilmainen xml-muokkain, joka eroaa radikaalisti perinteisestä Notepadista. XML Notepadilla voit nimittäin tuottaa koodia graafisen käyttöliittymän avulla." (Ks <http://www.mikrobitti.fi/nettijatkot/2002/08/xml-ohjelmointi/>)



Tehtävä

Tee XML-dokumentti käyttäen XML Notepadia.
Aiheena on osoitekirja

21.17 XSL

XSL :)

XSL eli Extensible Style Language on standardi, jonka avulla XML-dokumentit voidaan muuttaa helppolukuisiksi tai tulostuskelpoisiksi. Se jakaantuu kahteen osaan: XSLT ja XSL-Fo. XSLT on kieli, jonka avulla XML dokumentti voidaan muokata toiseksi rakenteiseksi dokumentiksi kuten esim. HTML:ksi. XSL-FO:n avulla XML-dokumentista voidaan laatia esitys- tai tulostuskelpoinen dokumentti kuten esim. pdf-tiedosto. (Kalliola 2001, 97.)

XSL on paljon enemmän kuin pelkkä ulkoasu, koska sillä voidaan mm määritellä järjestys. Voit esim. sen avulla laittaa CD-levysi aakkosjärjestykseen.

XML-dokumentti

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="catalog.xsl"?>
<luettelo>
```

```
<cd>
<nimi>Sticker hårt</nimi>
<artisti>Markoolio</artisti>
<maa>Ruotsi</maa>
<hintaa>21</hintaa>
<vuosi>1998</vuosi>
</cd>
```

```
<cd>
<nimi>Helvetisti järkee</nimi>
<artisti>Zen Cafe</artisti>
<maa>Suomi</maa>
<hintaa>2001</hintaa>
```

```

<vuosi>1982</vuosi>
</cd>
<cd>
<nimi>Popmuseo</nimi>
<artisti>Neljä ruusua</artisti>
<maa>Suomi</maa>
<hintaa>16.90</hintaa>
<vuosi>2000</vuosi>
</cd>
</luettelo>

```

XSL-dokumentti

```

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

<xsl:template match="/">
<html>
<head>
<title>CD-kokoelma</title>
</head>
<body>
<table border="1">
<xsl:apply-templates select="luettelo/cd">
<xsl:sort select="artisti" order="ascending"/>
</xsl:apply-templates>
</table>
</body>
</html>
</xsl:template>

<xsl:template match="luettelo/cd">
<tr>
<td><xsl:value-of select="artisti"/></td>
<td><xsl:value-of select="nimi"/></td>
</tr>
</xsl:template>

</xsl:stylesheet>

```

Mitä tapahtuu ja mitä pitää muistaa?

1) Muista liittää xsl XML dokumenttiin

```

<?xml-stylesheet type="text/xsl" href="catalog.xsl"?>

```

2) XSL kääntää XML:n HTML:ksi

```

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

```

3) XSL sisältää HTML-koodia

```

<html>
<head>
<title>CD-kokoelma</title>

```

</head>

4) Tiedon hakeminen ja sen järjestäminen

```
<xsl:apply-templates select="luettelo/cd">
<xsl:sort select="artisti" order="ascending"/>
```

Tehtävä

Harjoittele. Laita mukaan taulukkoon hinta ja maa ja vaihda lajittelujärjestystä.

22.18 Islandit

Islandit

Data Islandien avulla voidaan html: sekaan hakea XML:ää., toimii ainakin Explorer selaimen uusilla versioilla. On helppo ja vaivaton tapa tehdä tavallisesti HTML-sivusta dynaaminen sivu, jonka sisältö muuttuu samalla kuin taustalla oleva XML-dokumentti muuttuu.

Esimerkki

```
<html>
<body>
<xml
src="testia.xml"
id="xmldso"
async="false">
</xml>
<table
datasrc="#xmldso"
width="100%"
border="1">
<thead>
<th>Title</th>
<th>Artist</th>
<th>Year</th>
</thead>
<tr align="left">
<td><span datafld="nimi"></span></td>
<td><span datafld="artisti"></span></td>
<td><span datafld="vuosi"></span></td>
</tr>
</table>
</body>
</html>
```

Toimii siis [näin](#).

Lisää esimerkkejä löydät [täältä](#).

23.19 XML-pohjaisia uutisia

19. XML-pohjaisia uutisia

Lyhenne Rss tulee sanoista Really Simple Syndication (tai Rich Site Summary). Rss on osa XML-perhettä, ja se on taipuisa monenlaiseen käyttöön. Järvisen (2004, 76) mukaan Rss on sateenvarjotermi, jonka alta löytyy useita samaan tarkoitukseen tehtyjä XML-pohjaisia määrittäjiä. Rss:n ajatuksena on tarjota tapa tallentaa ja julkistaa uutiseksi katsottava tieto siten, että se on koneellisesti käsiteltävissä. Uutiseksi kelpaa mikä tahansa, se voi olla maailmanluokan uutinen tai sitten tieto siitä, että kotisivua on päivitetty.

Järvisen (2004, 76) mukaan Rss:n vahvuuksia on ainakin kaksi. Ensinnäkin Rss vapauttaa ihmiset siitä, että sivuilla vierailtaisiin jatkuvasti. Enää ei siis tarvitse surffata sivulle katsomaan, onko siellä jotakin uutta. Uutiset välittää kätevästi Rss. Voit siis mennä vierailemaan sivuille, mutta vasta sitten kun siellä on jotakin uutta. Toinen rss:n vahvuus on, että tieto välittyy paremmin perille kuin sähköpostitse. Kuten viime aikoina on nähty niin sähköpostilistojen kautta lähetetyt postit saattavat jäädä palveluntarjoajan filttereihin. Lisäksi oma sähköpostilaatikko pysyy siistinä.

Tehtävä

Tee itsellesi rss-sivu

Ohjelmia:

[Rssreader](#)

Validaattori

<http://feedvalidator.org/>

Katso esimerkki [Rss-sivusta](#).

Tehtävä

Tee oma Rss-sivu

Muista validoida se.

24.20 Tehtävä

20 Tehtävä

Etätehtävä, joka on palautettava ehdottomasti viimeistään 29.2.2004!

Tallenna XML dokumnetti Errarelle (ilmoita milloin valmis ja mistä löytyy) tai lähetä koko paketti minulle s-postitse, osoite jari.sarja@internetix.fi

Katso [html-sivua](#).

Tee sivusta oma XML versiosi. Voi itse päättää millainen ulkoasu. Tuleeko kuvia mukaan tai tuleeko linkkejä. (Kuvat ja linkit eivät ole pakollisia ottaa mukaan, mutta niistä saa lisäpisteitä).

Omaksi iloksi voi testailla osaamistasi [ristikolla](#).

25.21 Arviointi

21 Arviointi

Kurssisuoritus arvioidaan viikon kuluessa laskien viimeisestä tehtävän palautuspäivästä. Numero ja palautearviointi käytössä.