Helsingin yliopisto
Tietojenkäsittelytieteen laitos
Digitaalisen median tekniikat, kurssikoe 15.10.2007
Harri Laine

Vastaa tehtäviin 1ja 2 samalla konseptilla ja tehtäviin 3 ja 4 toisella konseptilla. Kirjoita kumpaankin konseptiin nimesi selvästi, syntymäaikasi tai opiskelijanumerosi, kurssin nimi (DIME) ja nimikirjoituksesi.

1. Selvitä miten määritellään CSS-säännön kohde (millaisia kohteita voidaan osoittaa) (6p)

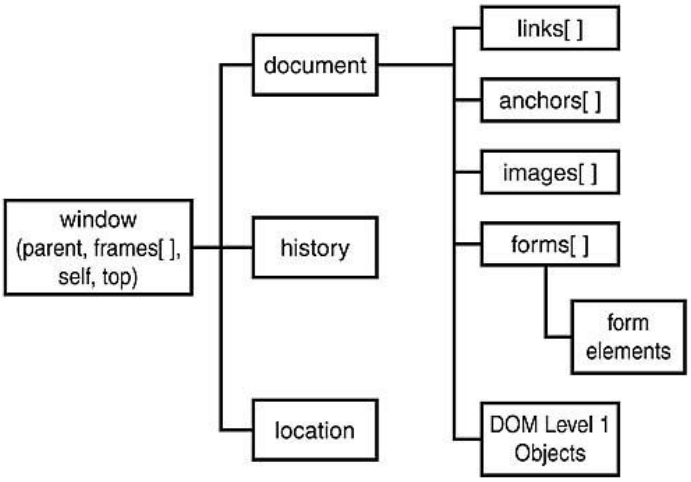
Kohteena voi olla:
- mikä tahansa elementtityyppi yleisesti, esim. **p** (tekstikappale) (1p)
- tunnisteella yksilöity elementti, esim. **#menu1** (1p)
- mikä tahansa tiettyyn luokkaan kuuluva elementti, esim **.sisalto** (0.5p)
- tiettyyn luokkaan kuuluva tietyn tyyppinen elementti esim **p.sisalto** (0.5p)
- kohde toisen kohteen sisällä hierarkkisessa rakenteessa esim. **p.sisalto a:visited** (1p)

- osalla elementeistä elementti tietyssä käsittelytilassa, esim **a:visited**
- joillakin elementeillä tietty elementin piirre esim, **p:first-line**
- **molemmat yllä = pseudoluokiteltu kohde**
- kohde toisen kohteen sisällä ylimmällä hierarkiatasolla esim **p.sisalto>a:visited**
- kohde joka on toisen kohteen välitön seuraaja peräkkäisjärjestyksessä , esim **p.sisalto+p**
- tietyn attribuutin sisältävä elementti, esim. **[width]**
- tietyn attribuuttiarvon sisältävä elementti , esim. **[width=100px]**
- tietyn tyyppinen tietyn attribuutin sisältävä elementti, esim **img[width]**
- tietyn tyyppinen tietyn attribuuttiarvon sisältävä elementti , esim.**img[width=100px]**
- kokoelma kohteita

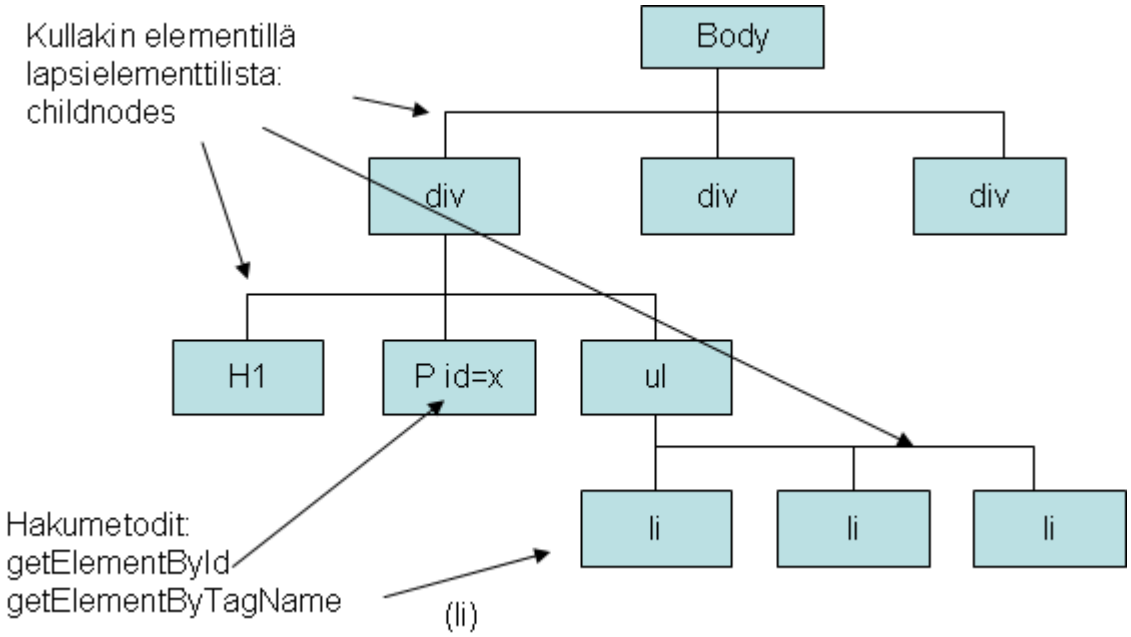
viivan alapuolella olevista 2 riittää yhteen pisteeseen.

2. Selvitä lyhyesti miten dokumentin sisältö näkyy Javascript-kielelle ja miten kielellä päästään käsittelemään dokumentin sisältöä tai ulkoasua. (6p)

Dokumentti näkyy olioista muodostuvana **puurakenteena**, ns. dokumenttipuuna (DOM). **Oliot vastaavat HTML:n elementtejä** ja **puurakenne vastaa dokumentin elementtien hierarkkista rakennetta**. Kullakin elementillä on joukko yhteisiä tietorakenteita mm tunniste, tyyli, luokitus, **lapsielementtien lista (ChildNodes) ja attribuuttilista**. Elementteihin voi osoittaa puun juuresta etenevällä piste-notaatiolla esim. document.form1.field1 käyttämällä elementtien nimiä. Kaikilla elementeillä ei kuitenkaan ole nimiä ja edellä esitetty tapa sopii lähinnä lomakkeiden käsittelyyn. Yleisesti elementtiin päästään käsiksi liittämällä dokumentissa **elementtiin tapahtumakäsittelijöitä** (esim. OnMouseover) ja välittämällä elementti käsittelijälle parametrina. Elementtejä voidaan valita operaatioiden kohteeksi myös käyttämällä elementtiolioiden tarjoamia **hakumetodeja** ja lapsielementtilistaa. **Hakumetodit mahdollistavat yksittäisen elementin haun tunnisteen perusteella (getElementByID) ja elementtikokoelman haun elementtityypin (tag) perusteella (getElementsByTagName).** Näiden perusteella löytyy käsittelylle aloituskohta, joista voidaan edetä lapsielementtilistaa tai hakumetodeja käyttäen eteenpäin.



Kuva1: Yleisrakenne



Kuva 2: esimerkki DOM Level -rakenteesta

Yllä olevassa kuvassa 2 dokumentin html-elementtejä (body, p, ul, form jne) vastaavia dokumenttipuun solmuja. Elementin välittömät lapsielementit muodostavat kyseisen elementin lapsielementtilistan. Elementeillä **on metodeja uusien lapsielementtien lisäämiseen. ja elementin attribuuttien arvojen muuttamiseen.**

Lihavoidut asiat tärkeitä periaatteen kannalta -metodien nimet eivät.

3. Selvitä lyhyesti mitä tarkoitetaan AJAX-tekniikalla ja miten tällä tekniikalla toteutettu sovellus eroaa perinteisestä web-sovelluksesta. Mitä etuja ja toisaalta haittoja tekniikan käytöllä on? (6p)

Perinteisessä web-sovelluksessa sivut tuotetaan kokonaan palvelimessa. Ne voivat olla staattisia tai ohjelmallisesti rakennettuja. Lähetettyään pyynnön palvelimelle selain jää odottamaan vastaussivun saapumista.

Ajax –tekniikalla sivut tuotetaan osittain palvelimessa ja osittain selaimessa. Selaimessa **dokumenttia muokataan Javascriptillä**. Javascriptillä voidaan myös lähettää palvelimelle pyyntöjä. Näiden pyyntöjen vastauksena ei toimiteta valmista HTML-sivua vaan aineistoa, jonka selaimessa toimiva Javascript-ohjelma sitten upottaa selaimessa esitettävään dokumenttiin.

AJAX tulee sanoista Asynchronous Javasrcipt and XML. Javascriptiä käytetään sivun muokkaukseen selaimessa ja palvelupyyntöjen lähettämiseen palvelimelle**. Javascript tarjoaa XMLHttpRequest-rajapintaolion (tai vastaavan komponentin) asynkronisten palvelupyyntöjen lähettämiseen.** Asynkronisella palvelulla tarkoitetaan sitä, että ohjelma ei jää odottamaan vastausta vaan ohjelmassa määritellään vastauskäsittelijät, jotka tapahtumaperustaisesti heräävät käsittelemään saapunutta vastausta. Vastaus voi olla tekstidataa tai XML-rakenteista tietoa. Palvelun tuottaja tai sovelluksen toteuttaja päättää missä muodossa vastaukset toimitetaan..

Ajaxin etuja: Verkossa siirrettävän datan määrä yleensä vähenee. Tietoa voidaan hakea taustalla. Tiedonhaut voidaan dynaamisesti sovittaa käyttäjän toimintaan. Käytettävyyttä voidaan parantaa uudenlaisilla käyttöliittymäelementeillä ja - toiminnoilla. Palvelimen ja asiakkaan kytkentää voidaan löyhentää, koska palvelimen ei tarvitse tietää mihin tietoa käytetään.

Haittoja: Edellyttää Javascriptiä, joka ei tältä osin toimi täysin yhdenmukaisesti eri selaimissa. Aiheuttaa esteettömyysongelmia. Tuottaa pimeää, hakukoneissa näkymätöntä tietoa. Voi lisätä tietokannan kuormitusta. Ratkaisut saattavat edellyttää nopeaa vasteaikaa toimiakseen hyvin.
Perusideasta 4p ja eduista ja haitoista 2p

4. Mitä tarkoitetaan webin pimeällä (tavoittamattomalla) tiedolla (invisible web data, deep web)? Anna esimerkkejä pimeästä tiedosta.(6p)

Pimeällä tiedolla tarkoitetaan tietoa, joka jää hakukoneiden ulottumattomiin. On arvioitu, että tällaista tietoa on yli 80% kaikesta webin kautta saavutettavissa olevasta tiedosta..Syitä tähän

- hakukoneiden hakurobotit eivät pääse tietoa esittävälle sivulle
- esim.
    - sivut, joille ei ole linkkejä
    - sivut jotka saadaan vastauksena jonkin avoimen hakuehdon tai lomakkeen täyttämiseen
    - sivut, jotka ovat käyttäjätunnuksen ja salasanan takana
    - sivut, joiden tutkiminen on kielletty
    - paikallistiedon perusteella rakennettavat kontekstisidonnaiset sivut
- hakurobotit eivät onnistu löytämään tietoa sivulta
- esim:
    - vaikeasti tulkittavat aineistot - kuvat, ääni, video
    - selaimessa dynaamisesti rakennettavat sivut
- sivu ei täytä indeksointiarvoisen sivun kriteerejä
- esim.
    - paljon rakennevirheitä, toimimattomia linkkejä,
    - viimeisestä päivityksestä liian kauan
- tieto on liian tuoretta

- hakuroboteilla on paljon tutkittavaa, voi kestää viikkoja ennen kuin sivusto tutkitaan ja indeksoidaan

Määritelmä 2p, loput 4 p esimerkeistä.