



Digitaalisen median tekniikat

Luento 6: Esteettömyys ja saavutettavuus



Luennot

1. Intro
2. XHTML
3. CSS
4. JavaScript
5. JavaScript-kirjastot & AJAX
6. **Esteettömyys & saavutettavuus**
 - Suosituksia
 - Esimerkkejä
7. Palvelinohjelmointi (PHP)
8. PHP jatkuu + pelkkiä kysymyksiä luento
9. Video- ja kuvaformaattit
10. Quirks
11. Hyviä, mutta vähän käytettyjä web-standardeja
12. Yhteenveto

Miksi?

- Sivuston kehittäjä ei saa päättää, miten tieto tulee katsoa
 - ”It’s their web – we are just building it.”
- Harvat web-sivut ovat oikeasti esteettömiä
- Tulevaisuudessa Internetin rooli tiedon jakamisessa kasvaa entisestään
- Esteettömyyden, saavutettavuuden ym. huomioiminen parantaa sivuston rakennetta ja täten myös käytettävyyttä
- Huomioitava koko projektin aikana, loppuvaiheessa voi olla mahdotonta
- Suuremman www-sivuston elinkaari voi olla hyvinkin pitkä, se pitää tehdä huolella
 - ...kotska-webskat ovat sitten asia erikseen

Määritelmiä

- Esteettömyys (accessibility)
 - Tieto saatavilla etenkin toimintarajoitteisille
 - Internetissä toimintarajoitteisuus ei pelkästään ihmisessä, vaan myös tekniikassa: yhteydet, selaimet...
- Käytettävyys (usability)
 - Tehokasta, miellyttävää käyttöä
 - Käytettävyys on laaja aihe ja kuuluu oikeastaan kurssille ”[Käyttöliittymät](#)”
 - Tällä kurssilla käytettävyyttä käsitellään esteettömyyden kautta
- Saavutettavuus (accessibility? reachability?)
 - Helppo lähestyttävyyys kaikille, osittainen synonyymi esteettömyyden kanssa
- Saatavuus, tavoitettavuus
 - Palvelu (tai tieto) on saatavilla kaikille
- Ymmärrettävyys (understandability)
 - Teksti on ymmärrettävää
 - kieli, terminologia
 - Demo: <http://s60.pipip.de/details.php?id=509>

WCAG suositukset

- Web Content Accessibility Guidelines
- W3-konsortion määrittely esteettömyydestä
- WCAG 1.0 1999
 - Ottaa huomioon eri toimintorajoitteet
 - Kolme prioriteettia noudattamiselle:
 - Must, should, may
- WCAG 2.0 2008
 - Tarkennettu määritelmiä, hienojakoisempi
 - Jokainen osa-alue yksiselitteisesti testattavissa

Hakurobotit esimerkkinä

- Hakurobotit (web crawlers, spiders, worms, ants...) perustuvat indekseihin joita ne päivittävät vieraillemalla säännöllisesti sivuilla.
- Hakurobotti on äärimmäinen esimerkki rajoitteisesta webin käyttäjästä
 - Robotti ei ”näe” sivua, eikä osaa päätellä yhteyksiä ilman algoritmeja
- Robotti aloittaa sivun selaamisen joltain aloitussivulta ja indeksoi haluamansa tiedot
- Indeksoinnin onnistumista voidaan parantaa
 - **hakukoneoptimointi** (SEO: Search Engine Optimization)
 - ei (yleensä) paranna esteettömyyttä ihmisille!
- Jos robotilla ei ole pääsyä johonkin osaan tiedosta, niin...

PIMEÄ WEB

Pimeä web ”meillä”

- Alma
 - Ei pääsyä ilman yliopiston tunnuksia
- Moodle
 - Ei pääsyä ilman kurssiavainta
- Laitoksen sivut
 - Rikkoutuneita linkkiketjuja, vanhentuneita dokumentteja uusien rinnalla
 - <http://www.cs.helsinki.fi/kurssit/aine/58168/>
 - PHP-ohjeet yms, yms...
- Ja muualla Internetissä vaikka mitä...

Pimeä web

- Sivulla käytetään jotain tekniikkaa
 - Jota ei ole asiakkaalla
 - Flash, JavaScript, Java...
 - Jonka tuottamaa sisältöä asiakas ei voi käyttää
 - Näytönlukija ei osaa lukea Flashillä tuotettua informaatiota
 - Eikä hakurobotti
- Linkittämätön tieto (solmut!)
- Hakukoneilta piilotettu tieto (robots.txt)
- Yksityinen tieto
 - Salasanat, IP-rajaukset
- Kontekstisidonnainen tieto
 - Tiedon esitys riippuu siitä mitä kautta sivulle on tultu

Valoa pimeyteen

- Vaihtoehtoiset esitysmuodot
 - Roboteille
 - Ihmisille
- Sivukarttojen käyttö
 - http://en.wikipedia.org/wiki/Site_map

Esteettömyys

- Aistivammat, aistien ”vajaa toiminta”
- Motoriikan häiriöt
- Myös ihmisen kärsimättömyys, kiire ja asenne?
- Ongelmia on vaikea nähdä koodissa, niiden löytämiseen tarvitaan apuvälineitä
 - Webaim wave: <http://wave.webaim.org/>
 - Firefox Accessibility Extension

Värit

- Värisokeus on todella yleistä
- Väriin ei saa luottaa ainoana visuaalisena vihjeenä
- Värien kontrasti huomioitava
- Visccheck:
<http://www.vischeck.com/vischeck/vischeckURL.php>

Teksti

- Koko oltava muutettavissa 200%, ilman että sisältö hajoaa lukukelvottomaksi
- Erikoisfontteja vältettävä
 - Jos on jostain syystä ihan pakko, niin kuvana (mielellään ei)
 - sIFR: <http://www.mikeindustries.com/blog/sifr/> (mielellään ei)
- Sanasto
 - Lyhenteet selitettävä tarvittaessa (<abbr>) → lisää ymmärrettävyyttä
- Linkit
 - Teksti kuten [Klikkaa tästä](#) ei kerro mitään linkistä.
 - Linkistä on käytävä ilmi, että se on linkki
- Kuvajaat voidaan piirtää myös ilman kuvia, säilyttäen samalla tekstimuotoinen tieto
 - Esimerkiksi: <http://cssglobe.com/post/1272/pure-css-data-chart>
- XHTML 2.0 (tulossa ihan kohta) selkeyttää rakennetta
 - <section>, <h>
 - <object> - Kuvat, äänet, videot ja vaihtoehtoiset esitystavat yhdessä

Teksti puheena: CSS3

- Tekstin oikea lausunta on myös sisällön ”ulkoasun esittämistä”
- Tämä on ihan yhtä tärkeää, kuin luettava kirjasintyyppi tai taustaväri.
- Demo:
<http://www.robodesign.ro/mihai/my-projects/css-3-speech-demo>

Taulukot

- Taulukot ovat hyviä ainoastaan taulukkomaisen datan esittämiseen
- Taulukoihin voi lisätä tietoa rakenteesta jotta näytönlukijat tietävät missä kohtaa taulukkoa liikutaan
 - scope, summary, header, axis
- <http://juicystudio.com/article/complextableinspector.php>

JavaScript

- Kuten JS-luennolla aihetta sivuttiin
 - Pystyykö sivustoa käyttämään ilman JavaScriptiä, onko kaikki sisältö saatavilla?
- href="#" tai href="javascript:..."
 - Sivusto ei varmasti toimi ilman
 - Jos toiminnon ei ole tarkoituskaan toimia ilman, niin elementti tulee myös luoda JavaScriptillä!
- Ajaxin käyttö oikein: älä tuhoa merkityksellistä historiaa
 - Oikea siirtymä aina, kun "sivu vaihtuu."

Rakenne ja esitys erilleen

- Taas! **Tämä on kurssin tärkein asia.**
- Rakenteellisesti vain looginen, jotta esitysmuoto voi vaihtua esitystavan mukaan.
- Rakenne
 - validia: tagit kiinni, id:t uniikkeja...
 - oikeaoppista
- Vältetään ongelmallisia toteutustapoja
 - Kehykset (frames) rikkovat webin osoitettavuuden ja rakenteen.

Navigointi

- Navigoinnin pääpiirteet eivät saa muuttua kesken selaamisen
- Hierarkiat selviä ja loogisia
- Otsikkojen käyttö oikein
- Leivänmurut
- Yleisten käytäntöjen noudattaminen

Siirtymät ja muu automatiikka

- Ajastettu siirtymä
 - `<meta http-equiv="refresh" content="5;url=uusi.html" />`
 - Tällainen siirtymä ei paranna käytettävyyttä
 - Oikea toteutustapa palvelimella ".htaccess" tai muulla tavalla
- "Autoplay" ilman kontrollia
 - Videoiden käynnistäminen, äänien soittaminen itsestään
- Pakotetut uudet ikkunat ja pop-upit
 - Päätelaitteesta ja käyttäjän tavasta selata sivua ei saa tehdä päätöksiä.
- Sisällön liikkuminen sivulla voitava estää
 - Skrollerit, marquee..

Tekniikat

- Tekniikkaa saa käyttää jos se on ”yleisesti käytössä”
- Koska uusi tekniikka ”yleisesti käytössä”
 - JavaScript?
 - Canvas?
 - CSS3 puhe?

Vaihtoehtoiset syötteet

- Näppäimistöt, ääniohjaus...
 - Accesskeys
 - <http://www.cs.tut.fi/~jkorpela/forms/accesskey.html#ex>
- Selainriippumattomuus
 - Noudattamalla standardeja mikä tahansa selain käy sivujen käyttöön
 - <http://www.digitoday.fi/tietoturva/2009/02/22/norjalaiset-tahtovat-eroon-ie-6sta/20094843/66>

CAPTCHA

following *finding*

- Completely Automated Public Turing test to tell Computers and Humans Apart.
- Kuvat ongelmallisia:
 - Koneen on kyllä vaikeahko lukea, mutta niin myös ihmisten
 - Luottavat (hyvään) näkökykyyn
 - Ääniversio?
- Päättelytehtävä vaihtoehtona...?
 - KittenAuth: <http://www.thepcspy.com/contact>

Latausnopeus

- Kurssin sivuilla luvattiin, että kerrotaan *”Palvelimen taakan siirtäminen asiakkaalle, sivujen optimointi”*
- Palvelimen taakan siirto:
 - Ei tehdä palvelimessa asioita, joita selain osaa jo tehdä.
 - **Suosituilla** sivustoilla voi olla tärkeää tuottaa sivu mahdollisimman nopeasti ja jättää sivun ”parantaminen” asiakkaalle
 - (Vähän huono esimerkki)
 - Sivulla lukee 100 kertaa ”kirjoitettu 3 tuntia 20 minuuttia sitten”
 - Tämä voidaan tehdä joko palvelimella:
 - » Kerrotaan suoraan dokumentissa
 - Tai asiakkaalla:
 - Palvelin kertoo vain aikaleiman ja asiakkaan JavaScript laskee aikaleiman ja nykyisen ajan välillä. Erotuksesta muodostetaan lause dynaamisesti.
 - » Aika voi päivittyä jatkuvasti, palvelimen muodostamana se ei voi
 - » Asiakaskoneen JavaScript tietää selaimen kielen ja voi lokalisoida sen mukaisesti

Latausnopeus

- Sivujen optimointi
 - Näillä on merkitystä käytännössä vain suosituilla ja ns. ”oikeilla” sivuilla
 - Pienennetään sivun HTML-rakenteen kokoa:
 - Käytetään keskitettyä CSS:ää, joka tarvitsee ladata vain kerran
 - Rakenne ja esitys erikseen (taas!)
 - Yhdistetään ulkopuoliset ”.js” JavaScript –tiedostot yhdeksi tiedostoksi
 - Pakataan sisältö
 - Ei liian suuria kuvia
 - Käytetään CSS:ää ja JavaScriptiä taustavärien ym. käyttämiseen → demo: <http://www.helsinki.fi/yliopisto/>
- Jatketaan palvelinohjelmoinnissa!

Kuvien alt -määreet

- Alt on vaihtoehtoinen esitystapa
 - Title on kuvateksti
- Jos kuva on ”koristeellinen”, niin alt=””

Lomakkeet

- Käsitellään nyt lomakkeet, niin voidaan hypätä PHP:n kanssa suoraan bileisiin!
- Taulukoita käytetään lomakkeiden asemointiin väärin
 - label, fieldset, legend