

7. Nettisivut ja ohjelmointi

Jaksossa opetetaan html-koodin perusteet, css-muotoilun perusteet ja ohjelmoinnin perusteet. Jakson alussa esitellään tehtäväkokonaisuus, jota lähdetään työstämään esimerkkien kautta.

Päivitetään vanhan oppimateriaalin tekstiä ja otetaan uudenlainen lähestymistapa HTML-koodin muokkaamiseen. Esim. WYSIWYG-editorit voidaan mainita, mutta niiden käyttöön uudessa materiaalissa ei keskitytä. Ns. online-editointisivut on hyvä pitää idealtaan samana uudessa materiaalissa, jolloin oppilas saa välittömän palautteen kirjoittamastaan koodista ja näkee muutokset heti. Oppilas lähtee työstämään koodia valmiilta pohjalta.

Oletetut esitiedot ennen jaksoa

Oppilas

- Ymmärtää ohjelmoinnin ja algoritmisen ajattelun perusteet.
- Oppilas on ohjelmoinut graafisessa ohjelmointiympäristössä.

Luvut ja lukukohtainen priorisointi

1. HTML perusteet (2 oppituntia)

a. www-sivustorakenne

b. elementit

c. muotoillut (siirretään CSS-muotoiluun)

2. CSS perusteet (1 oppitunti)

a. keskeisimmät muotoilut

3. JavaScript perusteet (3 oppituntia)

Jakson läpikäyminen vie yhteensä 6 oppituntia.

Jakson aikana tehtävä tehtäväkokonaisuus:

Oppilas saa aikaiseksi jakson päätteeksi internet-sivujen kokonaisuuden, joka käsittelee häntä itseään ja omia mielenkiinnon kohteita. Jakson sisältö linkitetään Teksinkäsittely-jaksoon.

Tavoitteet

Tämän jakson jälkeen

- osaat luoda itsenäisesti yksinkertaisia html-sivuja validilla koodilla
 - ymmärtää html-koodin syntaksin: aloitus- ja lopetustagi, attribuutit, elementit
 - ymmärrät divien käyttämisen perusidean
- osaat muokata ulkoasua yhden ulkoisen css-tiedoston avulla
 - ymmärtää css:n syntaksin
- pystyt tuottamaan yksinkertaisen javascript-komennon ja muokkaamaan sivuja javascriptiä käyttäen

1. HTML-perusteet (2 oppituntia)

Tavoitteet

Tämän luvun jälkeen

- osaat luoda itsenäisesti yksinkertaisia html-sivuja validilla koodilla
 - ymmärrät html-koodin syntaksin: aloitus- ja lopetustagi, attribuutit, elementit
 - ymmärrät divien käyttämisen perusidean

Johdanto (otsikkoa ei tekstiin)

Internetin käytetyin osa, World Wide Web (WWW), koostuu WWW-sivuista. WWW-sivut eivät ole vaikeita tehdä, helpoimmillaan niitä voi tehdä lähes tekstinkäsittelyn tavoin. Kuitenkin tässä luvussa opiskellaan periaatteet "ammattimaisemmasta" WWW-julkaisemisesta, eli WWW-sivujen tekemisestä koodia kirjoittamalla. Itse tekemiään WWW-sivuja voi katsoa omalta tietokoneelta. Jotta sivut näkyisivät Internetissä, tarvitaan palvelintilaa WWW-sivuille, jota tarjoavat useat firmat, muun muassa teleoperaattorit. Nykyään useat firmat (esimerkiksi Nettihotelli.fi) tarjoavat melko edullisesti myös nettihotellipalveluita. Niissä saa usein WWW-sivujen palvelintilan lisäksi sivut omaan domainiin, jolloin sivujen WWW-osoitteeksi voi valita esimerkiksi www.etunimisukunimi.net.

Tässä jaksossa puhutaan kolmesta käsitteestä: HTML:stä, CSS:stä ja JavaScriptistä. HTML (HyperText Markup Language) on nettisivujen rakennustyökalu. CSS (Cascading Style Sheet) on taas tyyliohjeiden määrittäjätyökalu. HTML:n avulla tehdään nettisivujen raamit, mutta CSS:n avulla raamit voidaan muotoilla ja asetella halutunlaiseksi. JavaScript on taas ohjelmointikieli, jolla voidaan lisätä toiminnallisuuksia HTML-sivuille.

HTML

HTML:n uusin versio on HTML5 ja tässä oppimateriaalissa käytetään myös sitä. Lisäksi on olemassa HTML:n laajennus, XHTML (The Extensible HyperText Markup Language). Molemmat ovat kuvauskieliä WWW-sivujen tuottamiseen. Suurin ero kielissä on se, että XHTML täyttää XML:n (Extensible Markup Language) muotovaatimukset ja on näin ollen hiukan monikäyttöisempi, mutta samalla myös standardeiltaan tiukempi.

HTML:n tiedostopäätte on .html. HTML-tiedostonimessä ei saa olla ääkkösiä, erikoismerkkejä tai välilyöntejä, sillä ne voivat aiheuttaa ongelmia sivujen selailussa joillakin selaimilla. WWW-sivuja voidaan katsella millä tahansa Internet-selaimella.

Yksinkertainen html-dokumentti

HTML:ää voi kirjoittaa tekstieditorilla (huom, ei tekstinkäsittelyohjelmalla). Windows-käyttöjärjestelmässä on valmiina Notepad ja Mac OsX-käyttöjärjestelmässä TextEdit (suom. TeXturi). Valmiiksi asennetuista ohjelmista on yleensä karsittu kaikki ylimääräiset ominaisuuden pois: syntaksiväriyty ja automaattinen sisennys ovat koodin kirjoittamista helpottavia ominaisuuksia, jotka ovat yleensä Internetistä on saatavissa ilmaisissa tekstieditoriohjelmissa (esim. Notepad++, Sublime Text, Vim, Atom).

Pakolliset ominaisuudet omaava peruspohja HTML-sivusta on seuraava:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Otsikko</title>
</head>
<body>

    <h1>Ensimmäisen tason otsikko</h1>
    <p>
        Tämä on tekstikappale. <br />
        Tähän tulee myös kuva myöhemmin.
    </p>

</body>
</html>
```

<kommentti>

(X)HTML-dokumentti koostuu tekstistä, jossa merkitään dokumentin rakenne elementeillä (tageilla), jotka kirjoitetaan < > -merkkien väliin. Esimerkiksi ensimmäisen tason otsikko on merkitty seuraavasti esimerkissä: <h1>. Elementeillä on aina aloitus- ja lopetustagi, joiden välissä on varsinainen elementin sisältö. Esimerkiksi edellä mainitun otsikon tapauksessa sisältö esitetään

seuraavasti: <h1>Ensimmäisen tason otsikko</h1>. <h1> on aloitustagi. Vastaavasti </h1> on lopetustagi. Lopetustagissa on aina /-merkki.

HTML-koodia kirjoittaessa on suositeltavaa, että omat blokit sisennetään omalle tasolleen (em. esimerkissä siis otsikko ja tekstiosio ovat sisennettynä, koska ne ovat samantasoisia osioita. Tämä tekee koodista helpommin luettavampaa.

Elementeillä voi olla ominaisuuksia eli *attribuutteja*, joilla määritellään elementtiin liittyviä ominaisuuksia. Esimerkiksi kuvan tapauksessa kuvan osoite, kuvan vaihtoehtoinen teksti ja kuvan otsikko määritellään img-elementtiin (kuva) tulevilla src (kuvan osoite), alt (kuvan vaihtoehtoinen teksti) ja title (kuvan otsikko) -attribuuteilla seuraavasti: . Attribuutin arvo siis syötetään lainausmerkkeihin = -merkin jälkeen. Ominaisuuksien lukumäärää ei ole rajoitettu. Yksi attribuutti, jota käsitellään tässä luvussa, on *class*, eli luokka. Luokka-attribuutin antamalla jokainen saman luokan omaava HTML-elementti saa saman muotoilun. Esimerkiksi tekstikappaleelle voidaan antaa luokka:

```
<p class="isompiteksti">Tämä teksti on kirjoitettu isommalla fontilla. </p>
```

Huomaa, että luokkanimeä julistaessa isoilla ja pienillä kirjaimilla on väliä. Siis "isompiteksti" ja "ISOMPITEKSTI" ja "IsompiTeksti" ovat kaikki eri luokkia. Luokka-attribuutti voidaan antaa mille tahansa HTML-elementille. Luokan muotoilu tapahtuu css:n avulla. CSS:stä kerrotaan seuraavassa luvussa.

Jos elementillä ei ole sisältöä, niin lopetustagi lyhennetään suoraan aloitustagiin. Esimerkiksi pakotetun rivinvaihdon tapauksessa onkin yllä merkitty:
. Elementtien välillä ja tekstin keskellä saa olla vapaasti välilyöntejä, sarkaimia ja rivinvaihtoja. Niillä ei ole merkitystä elementin sisällön ulkoasuun. Jos siis esimerkiksi halutaan tekstiin enemmän kuin yksi välilyönti peräkkäin, täytyy ylimääräiset välilyönnit koodata erikoismerkeillä (esimerkiksi välilyönnin koodi on);).

Avaa tekstieditori ja liitä sinne edellinen html-dokumenttipohja. Tallenna tiedosto nimellä esimerkki.html (paikkaan X). Tallennettuasi tiedoston, voit avata sen selaimella.

<kuva1>

<kuva2>

<tehtävä>

Nyt voit muokata html-tiedostoasi <title> -tagin (selaimen työkalupalkissa näkyvä dokumentin nimi), otsikon ja tekstikappaleen osalta. Täydennä esittelyteksti itsestäsi.

</tehtävä>

Divit

Termi *div* tulee sanasta division (osa / osa-alue) ja div-tageja käyttämällä html-sivu voidaan jakaa erilaisiin osiin, joiden muotoilua säädetään CSS:n avulla. Div-elementtejä voi sisällyttää myös sisäkkäin.

<kuva3>

Esimerkki diveistä:

```
<div class="sisalto">
  <h3>Otsikko</h3>
  <p>Ja tähän tulee tekstiä.</p>
</div>
<div class="sivupalkki">
  <h3>Toinen otsikko</h3>
  <p>Ja tähän tulee myös tekstiä.</p>
</div>
```

<huomiolaatikko>Ääkköset eivät ole suositeltavia luokkien nimissä</huomiolaatikko>

<tehtävä>

Luo neljä diviä, joilla on luokat navigointi, teksti, kuvat ja alaviite.

Navigointi: luo lista, johon teet kohdiksi "Esittely", "Koulutus" ja "Vapaa-ajan harrastukset"

<huomiolaatikko?>Numeroimattoman listan luonti käy seuraavasti:

```
<ul>
  <li>Ensimmäinen kohta</li>
  <li>Toinen kohta</li>
</ul>
</huomiolaatikko?>
```

Teksti: Kirjoita tähän vapaamuotoinen tervehdys kotisivuillesi ja mitä kotisivuiltasi löytyy.

Alaviite: Kirjoita tähän © ja oma nimesi, lisäksi vuosiluku.

</tehtävä>

Seuraavaksi lisätään css-muotoiluja sivulle.

Kuvat

Kuva 1: html-tiedosto kansiossa

kuva 2: html-tiedosto selaimessa

kuva 3: div-elementtien jaottelu sivulla

Huomiolaatikat

ääkköset

- a. www-sivustorakenne
- b. elementit

2. Css-perusteet (1 oppitunti)

CSS tulee sanoista Cascading Style Sheets, ja sillä muotoillaan html-elementtejä. CSS:n syntaksi on helppo:

```
ominaisuus: sääntö;
```

Mikäli css:ää upotettaisiin HTML:ään, se tulisi muodossa

```
<h1 style="ominaisuus: sääntö;">Ensimmäisen tason otsikko</h1>
```

ja upotuksia voisi tehdä useamman:

```
<h1 style="ominaisuus: sääntö; ominaisuus2:
sääntö;">Ensimmäisen tason otsikko</h1>
```

Tämä ei ole kuitenkaan kätevää **yleistämisen** takia, joten CSS-muotoilut kannattaa tehdä ulkoiseen css-tiedostoon. Csstiedoston päätte on .css .

Yksinkertainen CSS-tiedosto

```
body{
  margin: 0;
  padding: 0;
}
p{
  font-family: Arial;
  color: #000;
  font-size: 12px;
  line-height: 1.7;
}
```

Body-elementin määrittelyt tekevät yleiset, globaalit muotoilut koko sivulle. Huomaa aaltosulut, joiden sisään muotoilut sijoitetaan. Csstiedostossa voi julistaa muotoiluja tarvittavan määrän, eli mikään elementti ei ole css-tiedostossa pakollinen.

Linkittäminen html-tiedostoon

CSS-tiedosto voidaan linkittää html-tiedostoon sijoittamalla seuraava koodi <head> -tagien väliin:


```
<link rel="stylesheet" type="text/css" href="tyyli.css">
```

Tee siis nyt oma css-tiedosto tekstieditorilla ja linkitä se html-tiedostoosi. Lisää yleiset muotoilut (body), muotoilut tekstikappaleille (p), eri otsikkotasojille (h1-h3).

```
<laatikko1>
```

Divien muotoilu

Divien muotoilu tapahtuu helpoiten css:llä.

Esimerkiksi <div class="sisalto"> -diviä voitaisiin muokata seuraavasti:

```
.sisalto{  
    float: left;  
    text-align: center;  
    max-width: 75%;  
}
```

Luokka-attribuutti ilmoitetaan css-tiedostossa laittamalla attribuutin arvon eteen piste.

Kuvat

Huomiolaatikat

Laatikko 1: Sopivia muotoiluja syntaksin kanssa

- taustaväri
 - fontin väri
 - reunat
 - fontin määrittely
 - leveys ja korkeus (?)
 - margin ja padding
 - text-align
 - letter-spacing (?)
-
- a. syntaksi
 - b. keskeisimmät muotoilut
 - i. expande -laatikoihin useampi?
 - c. useamman sivun muotoilu yhdestä sivusta

3. JavaScript-perusteet (3 oppituntia)