

Verkkosivustojen suunnittelu ja rakentaminen

CSS-perusteet

Sisällysluettelo

- [1 Johdanto](#)
- [2 CSS:n syntaksi](#)
- [3 Valitsimet](#)
- [4 CSS:n liittäminen dokumentteihin](#)
 - [4.1 CSS erillisenä tiedostona](#)
 - [4.2 CSS upotettuna dokumenttiin](#)
 - [4.3 CSS:n sisällyttäminen HTML-elementteihin](#)
 - [4.4 Useampi CSS-määrittely samassa dokumentissa](#)
- [5 Periytyminen](#)
- [6 CSS:n ominaisuudet](#)
 - [6.1 Taustan tyylimäärittely](#)
 - [6.2 Tekstin tyylimäärittely](#)
 - [6.3 Taulukoiden tyylimäärittely](#)
- [7 Laatikkomalli](#)
- [8 Sivun taittaminen CSS:n avulla](#)
 - [8.1 Osa 1: yksinkertaisen sivun luominen](#)
 - [8.2 Osa 2: sivun elementtien asemointi](#)
 - [8.3 Osa 3: sivun viimeistely](#)
 - [8.4 Osa 4: CSS:n validointi](#)
- [9 Lopputehtävä](#)
- [10 Linkkejä](#)

© Otavan Opisto / Jari Sarja, Mikko Tarmia 2011



© Otavan Opisto / Jari Sarja, Mikko Tarmia 2011



1 Johdanto

Tervetuloa opiskelemaan CSS:n perusteita! CSS eli Cascading Style Sheets on erityisesti web-sivuille kehitetty tyylikieli, jonka avulla voidaan määrittellä HTML-, XHTML- sekä XML-dokumenttien ulkoasu ja esitystapa.

CSS-kieli kehitettiin erottelemaan dokumenttien sisällölliset ja ulkoasulliset tekijät toisistaan. CSS:llä on huomattavasti tehokkaampi kuvailla dokumentin ulkoasua kuin HTML- tai XHTML-kielillä. CSS:n etuja ovat mm.

- *Helpompi päivitettävyyys*, koska päivitettäviä tiedostoja on usein vain yksi.
- Kun kaikki turhat tyylimäärittelyt karsitaan pois HTML-/XHTML-dokumenteista, saavutetaan *pienemmät tiedostokoot* sekä *nopeampi latautuvuus*.
- *Parempi saavutettavuus ja laajempi esitysmediatuki* (mm. näkövammaisille). CSS:n avulla voidaan tehdä sivuista erilaiset versiot eri medioille.
- Entistä *parempi typografian kontrollointi*. CSS:n avulla voidaan vaikuttaa kaikkeen, millä on typografista merkitystä.

Tällä kurssilla opit rakentamaan yksinkertaiset web-sivut CSS:ää käyttäen. Kurssilla käydään aluksi läpi kielen perussäännöstö sekä keskeiset tyylimäärittelyt. Lopuksi taitetaan toimiva www-sivu. Kurssi vaatii HTML-kielen perusteiden hallintaa.

Voit käyttää sivujen tekemisessä vaikkapa Windowsin muistiota, mutta kehittyneemmän editorin käyttäminen helpottaa työskentelyä. Tällainen on esimerkiksi ilmaiseksi ladattava [Notepad++](#). Muita ilmaisia editoreja ovat mm. [NoteTab Light](#) sekä [EditPad Lite](#) (ei kaupalliseen käyttöön).

2 CSS:n syntaksi

CSS:n perussyntaksiin eli sääntöön kuuluu kaksi osaa: valitsin sekä yksi tai useampi määrittely. Valitsin on useimmiten HTML-elementti, jolle tyyli halutaan luoda. Määrittely puolestaan koostuu yhdestä tai useammasta ominaisuudesta (eli CSS-kielen tyylimääreestä) ja niiden arvosta.

Valitsimen ominaisuudet sijoitetaan aina aaltosulkujen sisään. Jokainen ominaisuus päätetään puolipisteeseen. Selkeyden lisäämiseksi ominaisuudet voidaan eritellä omille riveillensä.

Esimerkki:

```
p {                               / valitsin
  color:green;                    / 1. ominaisuus:arvo;
  text-align:center;             / 2. ominaisuus:arvo;
}
```

Määrittelyjen sekaan voidaan halutessa kirjoittaa myös kommentteja, joita käytetään selittämään kirjoitettua CSS-kieltä. Kommentit ovat selaimissa näkymättömiä eli ne eivät näy web-sivuilla.

```
/*
Kommentti kirjoitetaan aina vinoviivan ja asteriskin (eli tähtimerkin)
sisään.
Kommentin voi sijoittaa dokumentissa mihin kohtaan tahansa.
*/
```

CSS:n ominaisuudet ja mittayksiköt ovat kaikki kirjasinkoosta riippumattomia. Myöskään välilyönneillä tai rivinvaihdoilla ei ole merkitystä kielen tulkittavuuteen.

3 Valitsimet

Valitsin (Selector) määrittelee mihin dokumentin elementtiin tai elementtiryhmään CSS-tyylimäärittelyt kohdistetaan. HTML-elementtien lisäksi valitsimille voidaan luoda omat tunnisteensa *id*- ja *class*-valitsimia käyttäen. Valitsimien nimeämisessä kannattaa tarkistaa, että jokainen tunniste on uniikki ja etteivät ne sisällä erikoismerkkejä. Tunnisteen nimi ei myöskään saa alkaa numerolla.

Id-valitsimella voidaan määritellä tyyli yksittäiselle elementille. *Id*-tunniste on aina yksilöllinen; se ei saa esiintyä HTML-sivulla kuin kerran. Valitsin sijoitetaan HTML-tagiin määreeksi, esim. `<p id="punaisettekstit">` ja CSS-osiossa sen nimen eteen sijoitetaan tunnisteeksi “#”-merkki.

Esimerkki:

```
#punaisettekstit {
  font-size:12px;
  color:red;
}
```

Class-valitsimella tyylimäärittely voidaan kohdistaa kerralla joukolle elementtejä. *Id*-valitsimesta poiketen *class*-valitsimen etutunnisteena on “.”-merkki. Seuraavassa esimerkissä ne HTML-elementit keskittyvät, joiden määreenä on *class="keskitetty"*.

```
.keskitetty {
  text-align:center;
}
```

Class-valitsimen vaikutus voidaan myös rajata suppeampiin elementtiryhmiin, kuten taulukon soluihin, sijoittamalla tunnisteen eteen solujen valitsin “td”.

```
td.keskitetty {
  text-align:center;
}
```

4 CSS:n liittäminen dokumentteihin

CSS-tyylimäärittely voidaan yhdistää HTML- / XHTML-dokumenttiin usealla eri tavalla. CSS-tyylimäärittely voidaan

- tehdä erilliseksi tiedostoksi
- upottaa osaksi HTML-/XHTML-dokumenttia
- sisällyttää yksittäisiin HTML-elementteihin

Tapoja voidaan myös käyttää keskenään sekaisin.

4.1 CSS erillisenä tiedostona

CSS-tyylimäärittelyn luominen erilliseksi tiedostoksi on järkevin valinta silloin, kun sama tyyli halutaan monistaa useammalle web-sivulle. Näin ollen koko sivustokokonaisuuden tyylin muuttaminen käy kerralla kätevästi ainoastaan yhtä tiedostoa muokkaamalla. CSS-dokumentti saadaan tuotua käyttöön web-sivuille head-osioon sijoitettavan link-tagin avulla.

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

Vaihtoehtoisesti CSS-dokumentti voidaan liittää importoimalla se sivuille:

```
<head>
  <style type="text/css">@import url("omasivu.css");</style>
</head>
```

Erillinen CSS-tiedosto tallennetaan muodossa .css ja sen ei tulisi sisältää yhtään HTML-tagia.

4.2 CSS upotettuna dokumenttiin

CSS:n upottaminen dokumenttiin on hyödyllistä etenkin silloin, kun jokaiselle web-sivulle halutaan omanlaisensa tyylimäärittely. CSS-osa sijoitetaan tällöin head-osion sisälle *style*-tagia käyttäen.

```
<head>
  <style type="text/css">
    body {background-color:black;}
    p {margin-left:10px;}
    #punainen {color:red;}
  </style>
</head>
```

4.3 CSS:n sisällyttäminen HTML-elementteihin

CSS-tyylimäärittely voidaan tehdä myös kirjoittamalla määrittelyt itse HTML-tagien sisälle, mutta tämä metodi ei ole järin käytännöllinen, sillä se alkaa jo muistuttaa kömpelöä HTML-pohjaista tapaa muotoilla sivun ulkoasu. Sille voi olla silti käyttöä yksittäisissä tapauksissa (mm. pienet poikkeukset). Tyylimäärittely kirjoitetaan HTML-tagin sisään *style*-määrettä käyttäen:

```
<p style="color:red; text-decoration:underline;">Tässä on punaista ja
alleviivattua tekstiä.</p>
```

4.4 Useampi CSS-määrittely samassa dokumentissa

Jos edellä läpikäytyjä CSS-liittämistapoja on käytetty keskenään sekaisin samaa valitsinta käyttäen, viimeisin tyylimäärittely perii ja ylittää pääsääntöisesti dokumentin (lukusuunnassa) kaikkien aiempien tyylimäärittelyjen arvot. Samalla kaikki tyylimäärittelyt yhdistyvät toisiinsa. Tärkeys- ja perintäjärjestys on käytännössä seuraavanlainen:

1. Selaimen oletusarvot
2. Ulkoinen CSS-dokumentti
3. Dokumenttiin upotettu CSS
4. HTML-tageihin sisällytetty CSS

Esimerkki:

Ulkoisessa CSS-tiedostossa valitsimelle h1 on määritelty seuraavat ominaisuudet ja arvot.

```
h1 {
  color:green;
  text-align:left;
  font-size:10pt;
}
```

Edellistä CSS-tiedostoa käyttävään HTML-dokumenttiin upotetussa CSS-tyylimäärittelyssä valitsimen h1 ominaisuudet saavat uudet arvot väriä lukuunottamatta.

```
h1 {  
  text-align:center;  
  font-size:20pt;  
}
```

Samassa dokumentissa on käytetty CSS-tyylimäärittelyä *h1*-tagin sisällä. Uusien arvojen lisäksi voimaan jäävät myös *h1*-valitsimen aiemmin perimät arvot värille ja sijoittelulle.

```
<h1 style="text-decoration:underline; font-size:15pt;">Otsikko</h1>
```

Näin ollen HTML-tageihin sisällytetyt CSS-tyylimäärittelyt jyräävät alleen kaikki dokumentissa aiemmin esiintyneet arvot. Samoin HTML-dokumenttiin upotettu CSS ylittää ulkoisen CSS-dokumentin arvot. Ne eivät kuitenkaan muuta mitään, mitä ei ylikirjoiteta. Selain ikäänkuin lukee dokumentin ylhäältä alas ja sen muistiin jää aina viimeiseksi kirjoitettu muutos. Jos ulkoinen CSS-dokumentti olisi sijoitettu HTML-dokumenttiin upotetun CSS-tyylimäärittelyn perään sivun *head*-osiossa, nousisivat upotetut tyylimäärittelyt tärkeyslistalla ulkoisen dokumentin ohitse.

5 Periytyminen

Periytymisellä tarkoitetaan hierarkisen ajattelumallin mukaista tilannetta, jossa ns. lapsielementit perivät ominaisuutensa aina vanhemmiltaan. HTML-/XHTML-dokumentissa "suvun vanhin" on käytännössä aina *html*-elementti, jonka sisälle kaikki muut elementit sijoittuvat. Sama pätee muidenkin elementtien välisiin sisäkkyyssuhteisiin ominaisuuksia perittäessä - poikkeuksena mainittakoon taulukot, jotka eivät peri ominaisuuksia muilta kuin toisilta taulukkoelementeiltä.

Esimerkki:

```
<body>
  <h1>Otsikko</h1>
  <p>Yksi kappale, jossa on <b>lihavoitua tekstiä, mutta myöskin
    <i>kursivoitua tekstiä</i> seassa</b>.</p>
</body>
```

Kuten esimerkistä voidaan todeta, kaikki muut elementit sijoittuvat *body*-elementin sisälle. Näin ollen sille annetut CSS-tyylimäärittelyt vaikuttaisivat kaikkiin elementteihin. *h1*-elementillä ei ole lapsia, mutta *p*-elementin sisältä löytyy *p*:n lapsi *b* sekä tämän sisältä lapsenlapsi *i*. Jos tälle sivulle luotaisiin seuraavanlainen tyylimäärittely,

```
body {color:red;}
p {color:green;}
b {color:blue;}
i, h1 {text-decoration:underline;}
```

niin HTML-pohjaisen muotoilun lisäksi

- otsikko (h1) olisi punainen,
- kappaleen (p) alkuosa vihreä,
- lihavoitu (b) ja kursivoitu (i) teksti sinisiä sekä
- niin kursivoitu teksti kuin otsikkokin alleviivattua.

6 CSS:n ominaisuudet

6.1 Taustan tyylimääritteet

Sivun ulkoasua määritettäessä lähdetään usein liikkeelle taustaan liittyvistä ominaisuuksista. Taustan ulkoasun ominaisuuksia voidaan määritellä sivun taustan lisäksi myös pienemmille alueille, kuten taulukoille tai jopa pelkälle tekstille (vain väri).

CSS-kielessä taustan ulkoasua voidaan muuttaa seuraavilla ominaisuuksilla:

- background-color / taustan väri
- background-image / taustakuva
- background-repeat / taustakuvan jatkuvuus
- background-attachment / taustakuvan liikkuvuus sisältöön nähden
- background-position / taustakuvan asemointi

Background-color

Elementin taustaväri voidaan ilmaista selaimelle kolmella eri tavalla:

- hexadesimaaleina - "#ff0000"
- RGB-arvona - "rgb(255,0,0)"
- nimeämällä väri - "red"

Esimerkkejä:

```
body {background-color:yellow;}
table {background-color:#00ff00;}
p {background-color:rgb(255,0,255);}
```

Background-image

Taustakuvan tuominen sivuille onnistuu *background-image* -ominaisuutta hyödyntäen. Taustakuvan tarkka sijainti on ilmoitettava lausekkeessa - tässä tapauksessa kuva sijaitsee kuvat-nimisen kansion alla.

```
body {background-image:url('kuvat/taustakuva.jpg');}
```

Background-repeat

Oletuksena taustakuva toistaa itseään aina molempiin suuntiin koko elementin alueen verran. *Background-repeat* -ominaisuudella voidaan määritellä toistosuunta tai se, toistetaanko kuvaa ollenkaan.

Jos kuva halutaan toistaa ainoastaan vaakasuunnassa, käytetään *repeat-x* -arvoa

```
body {
  background-image:url('kuvat/taustakuva.jpg');
  background-repeat:repeat-x;
}
```

ja pystysuunnassa *repeat-y* -arvoa. Toistamisen voi estää *no-repeat* -arvon avulla.

Background-attachment

Taustakuvan voi suunnitella myös kiinteään kokoon ja asemaan siten, että se ei liiku vierityspalkkia siirrettäessä. Tätä varten on olemassa ominaisuus *background-attachment*, jolla on kaksi eri arvoa - *scroll* tai *fixed* (jälkimmäinen pitää kuvan paikallaan).

```
body {
  background-image:url('kuvat/taustakuva.jpg');
  background-repeat:no-repeat;
  background-attachment:fixed;
}
```

Background-position

Background-position -ominaisuudella saadaan määriteltyä taustakuvan aloitussijainti kohdelementin sisällä. Vaihtoehtoisia sijoitusarvoja:

```
background-position:center bottom; / suunta-arvot (left, right, top,
                                     bottom ja center sekä näiden
                                     yhdistelmät)
background-position:50% 0%; / %-arvo x/y (tässä: ylhäällä
                                     keskellä)
background-position:50px 100px; / x- ja y-arvot (tässä: 50 pikseliä
                                     vasemmalta, 100 ylhäältä)
```

Esimerkki:

```
body {
  background-image:url('kuvat/taustakuva.jpg');
  background-repeat:no-repeat;
  background-attachment:fixed;
  background-position:top center;
}
```

A01 tehtävä: Tee HTML-sivu, jonka tausta on muuten musta, mutta sivulla on myös keskitetty, pystysuunnassa jatkuva taustakuva. Kuva voi olla mikä tahansa, eikä sitä tarvitse lähettää tehtävän mukana. Käytä CSS-tyylimäärittelyä ja lisää se dokumenttiin upottamalla.

Jos suoritat tätä kurssia Muikussa, palauta tehtävätiedosto Muikun tehtävien kautta.

6.2 Tekstin tyylimääritteet

Tässä osiossa tarkastellaan tekstin tyylimääritteitä ja käydään läpi yleisimmät tekstin muotoiluun käytettävät CSS-ominaisuudet. Näitä ovat mm.

- font-family
- font-size
- font-style
- font-weight
- color
- text-align
- text-decoration

Lisäksi tässä osiossa tutustutaan myös listojen käyttämiin ominaisuuksiin.

Font-family

Kirjaisimet jakautuvat muotoilunsa mukaan viiteen eri kirjaisinperheeseen, joita ovat *serif* (esim. *Times*), *sans-serif* (esim. *Verdana*), *monospace* (esim. *Courier*), *cursive* ja *fantasy*. *Font-family*-ominaisuutta määritettäessä kannattaa mainita kirjaisimen lisäksi myös sen kirjaisinperhe, jotta selaimet osaisivat valita sen tilalle mahdollisimman samantapaisen kirjaisimen tapauksessa, jossa selain ei tue kyseistä kirjaisinta.

Ensimmäiseksi *font-family* -ominaisuuteen kirjoitetaan siis itse kirjaisin/kirjaisimet ja vasta viimeisenä varalle kirjaisinperhe. Jos kirjaisimen nimi koostuu useammasta sanasta, kirjoitetaan se kirjaisimen nimi lainausmerkkien sisään, esim.

```
p {font-family:"Times New Roman", Times, serif;}  
    / kirjaisin, varakirjaisin, kirjaisinperhe
```

Font-size

Kirjaisimen koon voi ilmaista monella eri tavalla, absoluuttisina tai suhteellisina arvoina. Absoluuttiset kirjaisinkoot varmistavat vastaavuuden eri selaimissa, mutta voivat aiheuttaa ongelmia mm. sivua pieninäyttöisillä laitteilla katseltaessa. Suhteellisilla arvoilla käyttäjä voi halutessaan itse säätää kirjaisinkoon selaimesta.

Kirjaisimen koko voidaan ilmaista mm.

pikseleinä:

```
p {font-size:14px;}
```

em-yksiköinä (varmistaa vastaavuuden myös Internet Explorerissa):

```
p {font-size:0.875em;}           / 1 em = 16 px, 0.875 em = 14 px
```

prosentteina:

```
p {font-size:100%;}           / = prosenttia peritystä kirjaisinkoosta
```

kirjoitettuina arvoina:

```
p {font-size: xx-large;}       / suhteessa perittyyn kirjaisinkokoon
```

```
p {font-size: smaller;}       / suhteessa perittyyn kirjaisinkokoon
```

Font-style

Font-style -ominaisuutta käytetään lähinnä tekstin kursivointiin.

```
p {font-style:normal;}         / normaali  
p {font-style:italic;}        / kursivoitu  
p {font-style:oblique;}       / kevyesti kallistunut
```

Font-weight

Font-weight -ominaisuutta käytetään puolestaan tekstin lihavointiin.

```
p {font-weight:normal}      / normaali
p {font-weight:bold}       / lihavoitu
p {font-weight:900}        / vapaa arvo 100-900, 400 = normaali
```

Color

Kuten taustavärienkin kohdalla, kirjaisimen väri voidaan ilmaista kolmella eri tavalla:

- hexadesimaaleina - "#ff0000"
- RGB-arvona - "rgb(255,0,0)"
- värin nimenä - "red"

Sivun kirjaisimien oletusväri ilmoitetaan *body*-valitsimessa.

Esimerkkejä:

```
body {color:blue;}
h3 {color:#00ff00;}
```

Text-align

Text-align -ominaisuus on yleispätevä keino eri elementtien sijoitteluun vaakasuunnassa. Teksti voidaan sijoittaa vasemmalle tai oikealle, keskittää tai tasata siten, että monirivisen tekstin molemmat laidat ovat keskenään suorassa (kuten lehdissä).

```
p {text-align:center;}      / keskittää tekstin
p {text-align:right;}      / sijoittaa tekstin oikealle
p {text-align:justify;}    / tasaa tekstin laidoilta
```

Text-decoration

Text-decoration -ominaisuudella voidaan korostaa tekstiä. Mahdollisia käyttötapoja ovat yli- ja alleviivaukset tai vaikkapa vilkkuva teksti.

Esimerkkejä:

```
p {text-decoration:none;}   / teksti ilman korostuksia
p {text-decoration:overline;} / ylleviivattu teksti
```

```
p {text-decoration:line-through;} / yliviivattu teksti
p {text-decoration:underline;} / alleviivattu teksti
p {text-decoration:blink;} / vilkkuva teksti
```

Listat

Siinä missä HTML-kieli tarjoaa hyvin suppean tavan tyylitellä listoja, CSS mahdollistaa uusien tyylimäärittelyjen lisäksi myös omien listakuvakkeiden käyttämisen. Tavalliset listakuvakkeet voidaan määrittellä *list-style-type* -ominaisuudella.

Esimerkkejä:

```
ul {list-style-type: circle;} / pallokuvake
ul {list-style-type: square;} / neliökuvake
ol {list-style-type: upper-roman;} / roomalaiset numerot
ol {list-style-type: lower-alpha;} / pienet aakkoset
```

Kuvan käyttäminen listakuvakkeena onnistuu *list-style-image* -ominaisuudella:

```
ul {list-style-image: url('kuvake.gif');}
```

Listan kuvakkeiden paikka sen aloitussijaintiin nähden voidaan määrittellä *list-style-position* -ominaisuudella:

```
ul {list-style-position:outside;} / vähentää sisennystä
ul {list-style-position:inside;} / lisää sisennystä
```

A02 tehtävä: Tee HTML-sivu, jolle muotoilet tarkasti alla olevan lyhyen tekstikatkelman CSS-tyylimäärittelyjen avulla. Valitse kirjaisimeksi Arial narrow, jolle varmistat myös mahdollisimman hyvän toimivuuden, mikäli selain ei tue sitä. Kirjaisinkoko otsikkoa lukuunottamatta on 16 pikseliä. Määrittele punainen "Cascading Style Sheets" -teksti vilkkuvaksi. Vaihda lisäksi pallokuvakelistan kuvakkeiksi neliöt ja numerolistan kuvakkeiksi suuret aakkoset. Lisää CSS-osio web-sivulle upottamalla ja tallenna dokumentti nimellä tekstintyylyttely.html.

Jos suoritat tätä kurssia Muikussa, palauta tehtävätiedosto Muikun tehtävien kautta.

CSS-TYLIKIELI

CSS eli **Cascading Style Sheets** on erityisesti web-sivuille kehitetty tyylikieli, jonka avulla voidaan määrittellä HTML-, XHTML- sekä XML-dokumenttien ulkoasu ja esitystapa.

CSS-kieli kehiteltiin erottelamaan dokumenttien **sisällölliset** ja **ulkoasulliset** tekijät toisistaan. CSS:llä on huomattavasti tehokkaampi kuvailla dokumentin ulkoasua kuin HTML- tai XHTML-kielillä.

CSS:n etuja ovat mm.

1. *Helpompi päivitettävyyys*, koska päivitettäviä tiedostoja on usein vain yksi.
2. Kun kaikki turhat tyylimäärittelyt karsitaan pois HTML-/XHTML-dokumenteista, saavutetaan *pienemmät tiedostokoot* sekä *nopeampi latautuvuus*.

Etuja ovat myös:

- *Parempi saavutettavuus* ja *laajempi esitysmediatuki* (mm. näkövammaisille). CSS:n avulla voidaan tehdä sivuista erilaiset versiot eri medioille.
- Entistä *parempi typografian kontrollointi*. CSS:n avulla voidaan vaikuttaa kaikkeen, millä on typografista merkitystä.

6.3 Taulukoiden tyylimääritteet

CSS:llä on helppo kohentaa HTML-taulukoiden ulkoasua. Osa taulukoihin käytettävistä ominaisuuksista onkin jo entuudestaan tuttuja, joten tässä osassa perehdytään niiden soveltamiseen sekä käydään läpi reunuksien ominaisuudet.

Taulukon koko ja reunukset

Kuten usean muunkin elementin kohdalla, taulukoiden ja solujen koon saa määriteltyä *width-* ja *height-*ominaisuuksien avulla. Samalla taulukolle voi luoda taustaväriin ja reunuksen käyttämällä *border-*ominaisuutta:

```
table {
  width:100%;
  background-color:green;
}
td {height:200px;}
table, td {border:1px solid black;} / maksimileveys, taustaväri vihreä
/ solujen korkeus on 200 pikseliä
/ taulukoilla ja soluilla on musta, kiinteä reunusviiva
```

Reunuksen määrittelyyn käytettiin sen lyhyempää muotoa *border*, joka voi sisältää kerralla enemmän arvoja reunuksen eri ominaisuuksien määrittelyyn. Saman lopputuloksen olisi saanut myös seuraavanlaisella, pidemmällä versiolla:

```
table, td {
  border-width: 1px;
  border-style: solid;
  border-color: black;
}
```

Reunusten tyylille on tarjolla eri vaihtoehtoja. Halutessa jokaiselle reunanpuoliskolle voidaan määrittellä oma tyylinsä, esimerkiksi seuraavalla tavalla:

```
td {
  border-top-style:solid;           / kiinteä reunus
  border-right-style:dotted;       / pisteellinen reunus
  border-bottom-style:dashed;      / katkoviivallinen reunus
  border-left-style:double;        / tuplaviivallinen reunus
}
```

Reunusviivan voi jättää myös kokonaan pois *none*-arvolla.

Selaimet luovat oletusarvona taulukon ja sen solujen reunusten väliin pienen tyhjän tilan. Reunukset voidaan yhdistää yhdeksi viivaksi *border-collapse* -ominaisuudella. Halutessa tätä tilaa voidaan myös kasvattaa *border-spacing* -ominaisuuden avulla.

```
table {border-collapse:collapse;} / yhdistää reunukset
table {border-collapse:separate;} / reunukset erillään, oletusarvo
table {border-spacing:10px;}     / taulukon ja solujen välinen tila
                                  on 10 pikseliä
```

Taulukkojen reunusta voi vielä tehostaa sen ulkopuolelle sijoittuvalla *outline*-ominaisuudella, joka on sijaintia lukuunottamatta identtinen *border*-ominaisuuden kanssa. Taulukon ulkopuolella olevaa tyhjää tilaa, ns. reunusaluetta, saadaan puolestaan muokattua käyttämällä CSS:n *margin*-ominaisuutta. *Margin* luo nimenomaan tyhjää tilaa, eikä sillä ole omaa taustaväriä.

Esimerkkejä:

```
table {outline:3px blue double;} / paksu sininen tuplareunusviiva
table {
  border:blue;
  outline:red;
} / punasininen reunusviiva
table {margin:10px;} / 10 pikselin mittainen reunusalue
```

Tekstin sijoittelu taulukossa

Tekstiä ja muuta sisältöä voidaan siirrellä melko vapaasti solujen sisällä. Jottei teksti sijoittuisi aivan kiinni reunukseen, tilan luomiseen voi käyttää *padding*-ominaisuutta (ns. täyte).

Esimerkkejä:

```
td {padding:10px;} / 10 pikselin tila tekstin ja
                        reunuksen välillä
td {padding:10cm 0cm 5cm 10cm;} / tila suunnissa
ylä/oikea/ala/vasen
                        ("kellon ympäri")
```

Tekstin sijoitteluun vaaka- ja pystysuunnassa voidaan käyttää *text-align*- ja *vertical-align*-ominaisuuksia. Seuraava esimerkki sijoittaa tekstin solun oikeaan alalaitaan:

```
td
{
  text-align:right; / vaakasuunta: left, right tai center
  vertical-align:bottom; / pystysuunta: top, middle tai bottom
}
```

Myös taulukon kuvaustekstin sijainti voidaan määritellä.

```
caption {caption-side:top;} / top, bottom, right tai left
```

A 03 tehtävä: Tee CSS:n tyylimäärittelyjä hyödyntäen kolmisarakkeinen taulukko, jossa on kentät asiakkaille, osoitteille ja puhelinnumeroille. Täytä tiedot ainakin muutaman keksityn asiakkaan osalta. Määrittele joka toisen asiakkaan rivi taustaväriksi valkoiseksi ja joka toinen vaaleansiniseksi. Anna otsikkosoluille sitä tummempi taustaväri. Tee taulukosta selkeästi luettava. Lisää CSS-osio web-sivulle upottamalla ja tallenna dokumentti nimellä asiakkaat.html.

Jos suoritat tätä kurssia Muikussa, palauta tehtävätiedosto Muikun tehtävien kautta.

7 Laatikkomalli

CSS-tyylimäärittelylle keskeisen “laatikkomalli”-käsitteen avulla kaikkia HTML-elementtejä voidaan ajatella laatikoina, jotka koostuvat neljästä toisiaan ympäröivästä osasta. Nämä ovat:

- sisältö (content) / laatikon varsinainen sisältö; teksti, kuvat jne.
- täyte (padding) / sisältöä ympäröivä tyhjä alue
- reunus (border) / sisältöä ja täytettä ympäröivä alue
- reunusalue (margin) / läpinäkyvä reunuksen ulkopuolinen alue



Laatikkomalli

Laatikkomalliin liittyviä ominaisuuksia käytiin läpi jo taulukoiden tyylimäärittelyä käsittelevässä osiossa 6.3. Samat ominaisuudet pätevät muihinkin elementteihin.

Elementin kokonaistilaa laskettaessa on muistettava ottaa huomioon kaikkien edellämainittujen osien vaatima tila. Esimerkiksi 400 x 300 pikselin kokoinen elementti voidaan muodostaa seuraavasti:

```
width: 360px;           / sisällölle varattu leveys
height: 250px;         / sisällölle varattu korkeus
padding:10px;          / yht. 20 pikseliä (10 px/suunta)
border:5px dotted black; / yht. 10 pikseliä (5 px/suunta)
margin:10px 5px 10px 5px; / korkeus yht. 20 px, leveys yht. 10 px
```

8 Sivun taittaminen CSS:n avulla

8.1 Osa 1: yksinkertaisen sivun luominen

Seuraavaksi luodaan toimiva web-sivu kurssilla käytettyjä CSS-tyylimääritteitä hyödyntäen. Liikkeelle lähdetään aivan yksinkertaisesta kaksipalstaisestä mallista, jossa vasemmalla puolella on valikko ja oikealla puolella varsinainen sisältö.

Aluksi tehdään sivun HTML-osio ilman CSS-tyylimääritteitä.

```
<html>
  <head>
    <title>Harjoitus</title>
  </head>

  <body>
    <ul>
      <li><a href="sivu1.html">Linkki sivuun 1</a></li>
      <li><a href="sivu2.html">Linkki sivuun 2</a></li>
      <li><a href="sivu3.html">Linkki sivuun 3</a></li>
    </ul>
    <h1>Otsikko</h1>
    <p>Tässä sijaitsee sivun varsinainen sisältö.</p>
  </body>
</html>
```

Tämän jälkeen elementit luodaan ja sijoitetaan laatikkomallin mukaisesti paikoilleen lisäämällä sivulle upotetut CSS-tyylimääritteet sekä identifioimalla valitsimet. Lopuksi laatikon ympärille lisätään vielä reunusviivat.

```
<html>
  <head>
    <style type="text/css">
      #navi {
        position: absolute;
        width: 150px;
        margin-left: 5px;
      }
      #sisalto {
        margin-left: 150px;
      }

      div {
```

```

        border:1px solid black;
    }
</style>
<title>Harjoitus</title>
</head>

<body>
  <div id="navi">
    <ul>
      <li>
        <a href="sivu1.html">Linkki sivuun 1</a>
      </li>
      <li>
        <a href="sivu2.html">Linkki sivuun 2</a>
      </li>
      <li>
        <a href="sivu3.html">Linkki sivuun 3</a>
      </li>
    </ul>
  </div>
  <div id="sisalto">
    <h1>Otsikko</h1>
    <p>Tässä sijaitsee sivun varsinainen sisältö.</p>
  </div>
</body>
</html>

```

Valikko ja sisältö jaettiin erilleen div-tageja käyttäen ("navi" ja "sisalto"). Linkkilistan sisältämä navi-laatikko on 150 pikseliä leveä ja sen reunuksen ulkopuolella on 5 pikselin reunusalue. Sisältö-laatikko on sijoitettu *margin*-ominaisuutta käyttäen. Laatikon absoluuttinen sijainti määriteltiin *position*-ominaisuudella.

Sivun kehittämistä jatketaan osassa 2.

8.2 Osa 2: sivun elementtien asemointi

Tässä osassa otetaan käyttöön lisää ominaisuuksia ja jatketaan samalla aiemmin aloitetun sivun työstämistä. Sivua laajennetaan seuraavalla tavalla:

```

<html>
  <head>
    <style type="text/css">
      body {
        text-align:center;
        padding:0px;

```

```

}
#isolaatikko {
  width:600px;
  margin:0px auto;
}
#navi {
  position:fixed;
  width:150px;
  margin-left:5px;
}
#otsikko, #sisalto {
  margin-left:150px;
}
#sisalto {
  height:100%;
  text-align:left;
  padding:20px;
}
div {
  border:1px solid black;
}
</style>
<title>Harjoitus</title>
</head>

<body>
<div id="isolaatikko">
  <div id="navi">
    <ul>
      <li><a href="sivu1.html">Linkki sivuun 1</a></li>
      <li><a href="sivu2.html">Linkki sivuun 2</a></li>
      <li><a href="sivu3.html">Linkki sivuun 3</a></li>
    </ul>
  </div>
  <div id="otsikko">
    <h1>Otsikko</h1>
  </div>
  <div id="sisalto">
    <p>Tässä sijaitsee sivun varsinainen sisältö.</p>
  </div>
</div>
</body>
</html>

```

Pienten laatikoiden ympärille tehtiin kokonaisuuden keskittämistä varten isompi laatikko (“isolaatikko”). Pelkkä *text-align* -ominaisuus *body*-elementin sisällä ei riitä laatikon keskittämiseen. Tämän vuoksi isolaatikko-elementtiin täytyi lisätä myös *margin*-ominaisuuden

vaakasuuntaiseksi arvoksi *auto* (0px kohdentuu pystysuuntaiseen arvoon). Näin laatikon ulkoreunus saa molemmille puolille täyden tilan keskittäen samalla itse laatikon. Navi-laatikon *position*-ominaisuuden status muutettiin tilaan *fixed*, joka estää navi-laatikon liikkumisen sivua alaspäin vieritettäessä. Lisäksi otsikko- ja sisalto-laatikot eroteltiin toisistaan. Sisalto-laatikko on sisällön määrän mukaan venyvä, mutta sille on annettu myös minimikorkeus 900 pikseliä. Huomaa, että otsikko perii keskityksensä *body*-elementiltä. Se sopii otsikon luonteelle, mutta sisältö on hyvä tasata laatikkonsa vasempaan laitaan.

Sivun kehittämistä jatketaan osassa 3.

8.3 Osa 3: sivun viimeistely

Aiemmin työstetty sivu viimeistellään tässä osassa. Sivulle tuodaan myös kuva, joka sijoitetaan dokumentin otsikko-kenttään.



```
<html>
  <head>
    <style type="text/css">
      body {
        text-align:center;
        padding:0px;
        background-color:#CC8866;
        font:14px Verdana, serif;
      }
      #isolaatikko {
        width:600px;
        margin:0px auto;
      }
      #navi {
        position:fixed;
        width:150px:
      }

      #otsikko, #sisalto {
        margin-left:150px;
        border:1px solid black;
      }
      #otsikko {
        height:80px;
```

```

        background-image:url('otsikkoteksti.jpg');
        background-repeat:no-repeat;
    }
    #sisalto {
        height:100%;
        min-height:900px;
        color:#ffffff;
        background-color:#000000;
        text-align:justify;
        padding:0px 25px 25px 25px;
    }
    a {
        color:#ffffff;
        text-decoration:underline;
    }
    ul {
        text-align:center;
        margin:0;
        padding:0;
    }
    li {
        color:black;
        width:150px;
        list-style:none;
        background-color:#333333;
        margin:0px 0px 2px 0px;
        padding:12px 0px;
        border:1px solid black;
    }
</style>
<title>Harjoitus</title>
</head>

<body>
<div id="isolaatikko">
<div id="navi">
<ul>
<li><a href="sivu1.html">Linkki sivuun 1</a></li>
<li><a href="sivu2.html">Linkki sivuun 2</a></li>
<li><a href="sivu3.html">Linkki sivuun 3</a></li>
</ul>
</div>
<div id="otsikko">
<h1>Otsikko</h1>
</div>
<div id="sisalto">
<p>Tässä sijaitsee sivun varsinainen sisältö. Tällä kertaa
tekstiä on hieman enemmän, jotta näkisimme justify-arvon
vaikutuksen.</p>

```

```
        </div>
    </div>
</body>

</html>
```

Yksi uudistuksista ovat vasemman palkin navigointipainikkeet, jotka on toteutettu lista-elementtien avulla. Jotta listan saisi mukavasti paikoilleen ilman reunoille ilmestyvää tyhjää tilaa, on *ul*-elementtiin määriteltävä *margin*- ja *padding*-ominaisuuksien arvoiksi 0. Muuten lista venähtää osittain otsikko-laatikon päälle. Tutki sivun tyylimääritteet tarkasti läpi, jotta ymmärrät mikä vaikuttaa mihinkin sivun ulkoasussa. Kokeile myös ominaisuuksien arvojen vaihtamista ja tutki muutosten vaikutuksia.

Seuraavassa osassa käydään läpi koodin validointi.

8.4 Osa 4: CSS:n validointi

Jotta kirjoitettu CSS-kieli olisi mahdollisimman puhdasta ja toimivaa, tehty dokumentti on syytä tarkistaa CSS-validaattorin avulla. Sitä ennen tutustutaan vielä HTML:n/XHTML:n doctype-määrittelyyn.

Doctype

DTD (Document Type Definition) on ohje selaimelle siitä, miten sen tulisi tulkita www-sivu. *Doctype* sijoitetaan aina www-dokumentin alkuun ja sen puuttuminen voi aiheuttaa sen, etteivät kaikki CSS-tyylimäärittelyt toimi oikein eri selaimilla (kuten Internet Explorerilla). Näin voi tapahtua myös, jos sivuston CSS-tyylimäärittelyt on tehty ulkoiseen dokumenttiin ja eri sivuilla on käytetty toisistaan poikkeavaa DTD-määrittelyä.

Käytössä on kolme erilaista *doctype*-määrittelyä:

Strict	/ Määrittelee vain rakenteen, ei ulkoasua. Ei frame-rakenteisille sivuille!
Transitional	/ Kuten strict, mutta antaa myös HTML:n vaikuttaa sivun ulkoasuun.
Frameset	/ Kuten Transitional, mutta mahdollistaa frame-rakenteen käytön.

Esimerkkejä:

```
Strict HTML:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

```
Transitional HTML:
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Frameset XHTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Aiemmin tällä kurssilla tehdyssä www-sivussa kaikki ulkoasuun liittyvät määritteet on tehty CSS-pohjaisesti ja HTML-kieltä on käytetty ainoastaan sivun rakenteelliseen puoleen. Määritteenä voi siis käyttää *strict HTML* -muotoa. Jos ulkoasun muotoiluun käytettäisiin CSS-tyylimääritteiden lisäksi myös HTML-kieltä, tulisi sivulle valita *Transitional*-muoto.

Doctype-rivi lisätään heti dokumentin alkuun. Sivu sisältää myös skandinaavisia merkkejä, joten head-osion sisään lisätään myös kaksi meta-riviä, jotka määrittelevät sivulla käytetyn merkistön sekä kielen. Merkistöksi valitaan UTF-8, joka takaa parhaan toimivuuden eri alustoilla. Kaikki erikoismerkit on aina enkoodattava ISO-8859-1-standardin mukaiseen ASCII-muotoon. Harjoituskirjoituksen kohdalla tämä tarkoittaa sitä, että kaikki ä- ja ö-kirjaimet muutetaan tekstieditorin *replace*-toiminnolla muotoon "ä" ja "ö". Tämä kannattaa tehdä vasta sivun sisällön kirjoittamisen jälkeen, jolloin kaikki merkit voidaan korvata yhdellä kertaa. Kattava lista merkistöstä löytyy mm. [täältä sivulta](#).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=UTF-8">
    <meta http-equiv="Content-Language" content="fi">
    <style type="text/css">
      body {
        text-align:center;
        padding:0px;
        background-color:#CC8866;
        font:14px Verdana, serif;
      }

      #isolaatikko {
        width:600px;
        margin:0px auto;
      }
      #navi {
        position:fixed;
        width:150px:
      }
      #otsikko, #sisalto {
        margin-left:150px;
        border:1px solid black;
```

```

}
#otsikko {
  height:80px;
  background-image:url('otsikkoteksti.jpg');
  background-repeat:no-repeat;
}
#sisalto {
  height:100%;
  min-height:900px;
  color:#ffffff;
  background-color:#000000;
  text-align:justify;
  padding:0px 25px 25px 25px;
}
a {
  color:#ffffff;
  text-decoration:underline;
}
ul {
  text-align:center;
  margin:0;
  padding:0;
}
li {
  color:black;
  width:150px;
  list-style:none;
  background-color:#333333;
  margin:0px 0px 2px 0px;
  padding:12px 0px;
  border:1px solid black;
}
</style>
<title>Harjoitus</title>
</head>

```

```

<body>
  <div id="isolaatikko">
    <div id="navi">
      <ul>
        <li><a href="sivu1.html">Linkki sivuun 1</a></li>
        <li><a href="sivu2.html">Linkki sivuun 2</a></li>
        <li><a href="sivu3.html">Linkki sivuun 3</a></li>
      </ul>
    </div>
    <div id="otsikko">
    </div>
    <div id="sisalto">

```

```
<p>T&auml;ss&auml; sijaitsee sivun varsinainen  
sis&auml;lt&ouml;. T&auml;ll&auml; kertaa teksti&auml; on  
hieman  
enemm&auml;n, jotta n&auml;kisimme justify-arvon  
vaikutuksen.</p>  
</div>  
</div>  
</body>  
  
</html>
```

CSS-validointi

Viimeisenä dokumentin CSS-tyylimääritteiden toimivuus tarkastetaan W3C:n sivuilta löytyvän [CSS-validaattorin](#) avulla. Tarkastettava sisältö voidaan syöttää validaattorille joko tekstikenttään syöttämällä, tiedostosta lataamalla tai suoraan verkossa sijaitsevasta osoitteesta.

Validaattori on välillä vähän turhankin tarkka ja se valittaa myös dokumentissa oikein olevista asioista, mikäli siitä puuttuu muita, sen mielestä hyvin olennaisia, määritteitä. Virhe-listalta voi välillä löytyä melko yllättäviäkin puutteita. Täysin oikeaoppisesti rakennettu sivu menee kuitenkin virheittä läpi, joten kannattaa tarkastaa validaattorin tuottama virhe-lista huolella lävitse.

9 Lopputehtävä

Kurssin lopputehtävässä mitataan oppineisuuttasi.

A04 tehtävä: Rakenna kurssilla oppimaasi tietoa hyödyntäen sivu, jossa navigointipalkit ovat laatikossaan vierekkäin sivun yläosaan sijoittuneina. Jaa sivun alaosa kahteen palstaan: sisältöön ja uutisiin. Muotoile sivu tyyllillä, mutta älä käytä kuvia. Lisää CSS-osio web-sivulle upottamalla ja tallenna dokumentti nimellä lopputehtava.html. Tarkasta sivu lopuksi validaattorilla.

Sivun rakenne on seuraavanlainen:



Jos suoritat tätä kurssia Muikussa, palauta tehtävätiedosto Muikun tehtävien kautta.

10 Linkejä

Tässä muutama hyödyllinen linkki kurssiin liittyen:

[World Wide Web Consortiumin CSS-validaattori](#)

[ISO-8951-1-merkistö](#)

[HTML-värikartta](#)

[Zengarden](#) - näyttäviä CSS:llä muotoiltuja sivukokonaisuuksia

[W3schools.com](#) - kaikenkattava CSS-opas

Editoreita:

[Notepad++](#)

[NoteTab Light](#)

[EditPad Lite](#) (ei kaupalliseen käyttöön)