

HTML-elementit

Tämä dokumentti sisältää useimmat [HTML 4.01](#) -suosituksen elementit attribuutteineen. Kirjoitin tämän lähinnä perusteelliseksi referenssiksi, en perinteiseksi oppaaksi. Mukana on silti runsaasti esimerkkejä, myös CSS-tyylisäännöistä. Dokumentti on jaettu kymmeneen lukuun, jotka näkyvät alla olevassa sisällysluettelossa.

Olen kirjoittanut yleisluontoista asiaa web-sivujen tekemisestä [Web-julkaiseminen](#)-sivulle sekä omat sivut web-sivuilla käytetyistä [merkistöistä](#) ja [väreistä](#). Katso myös [selaintestisivua](#).

Vanhentuneet (*deprecated*) elementit ja attribuutit on merkitty **punaisella** kommentilla, samoin uudet elementit jotka tulivat mukaan HTML 4 -suosituksessa. (Jos selaimesi ei tue CSS:ää, kommentit näkyvät kursivoituina.) Vanhentuneet elementit ja attribuutit olivat mukana [HTML 3.2](#) -suosituksessa. Ne ovat edelleen mukana uudemmassa HTML 4 Transitionalissa, tosin ”ei-suositeltuina”. HTML 4 Strict ei sisällä niitä ollenkaan.

Olen tässä dokumentissa jakanut elementit loogisiin ryhmiin. Mukana ei ole kaikkea mahdollista. Kehyksiin (frames) ja lomakkeisiin liittyvät jutut ovat toistaiseksi kirjoittamatta. Käytän termejä lohkoelementti ja tekstielementti—luvussa [Huomautuksia](#) kerrotaan mikä kuuluu mihinkin, joten haluat ehkä lukea sen ensimmäiseksi.

Tämä opas on tarkoitettu kirjoittajille, jotka jo hallitsevat HTML:n alkeet ja tarvitsevat tiivistä, selkeää referenssiä. Jos haluat aloittaa perusteista, kokeile vaikka Jukka Korpelan sivuja [Web-julkaisemisen opas](#) ja [Getting Started with HTML](#).

Sisällys

- ◆ **Ylimmän tason elementit:** [HTML](#), [HEAD](#), [BODY](#)
- ◆ **HEAD-elementit:** [TITLE](#), [META](#), [STYLE](#), [SCRIPT](#) ja [LINK](#)
- ◆ **Geneeriset lohkoelementit:** [H1...H6](#), [P](#), [DIV](#), [BLOCKQUOTE](#), [PRE](#), [HR](#), [ADDRESS](#), [NOSCRIPT](#), [CENTER](#)
- ◆ **Tekstielementit:** [A](#), [BR](#), [IMG](#), [MAP](#), [AREA](#), [SCRIPT](#), [SPAN](#), [Q](#), [OBJECT](#), [PARAM](#), [APPLET](#), [FONT](#)
- ◆ **Fraasit:** [EM](#), [STRONG](#), [CITE](#), [CODE](#), [KBD](#), [SAMP](#), [VAR](#), [ABBR](#) ja [ACRONYM](#)
- ◆ **Fonttityylit:** [TT](#), [B](#), [I](#), [BIG](#) ja [SMALL](#), [SUB](#), [SUP](#), [U](#), [S](#) ja [STRIKE](#)
- ◆ **Listat:** [UL](#), [OL](#), [LI](#), [DL](#), [DT](#) ja [DD](#), [DIR](#) ja [MENU](#)
- ◆ **Taulukot:** [TABLE](#), [CAPTION](#), [TR](#), [TD](#), [TH](#), [COL](#), [COLGROUP](#), [THEAD](#), [TBODY](#) ja [TFOOT](#)
- ◆ **Yhteiset attribuutit ja symboliset notaatiot**
- ◆ **Huomautuksia**

Markku Immonen <wired.poet@sci.fi>

Viimeksi muokattu: 17.2.2006

URL: <http://www.saunalahti.fi/wpoet/fin/htmlref.htm>

Ylimmän tason elementit

HTML

Dokumentti on aina kahden **HTML**-tagin välissä. Aloitustagi **<HTML>** tulee heti dokumenttityypin jälkeen, sen jälkeen tulevat **HEAD** ja **BODY**. **HEAD** sisältää metainfoa joka ei näy sivulla, **BODY** sisältää sivun varsinaisen sisällön. Aivan ensimmäisenä kerrotaan dokumentin tyyppi: HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset tai vanhempi HTML 3.2. Dokumenttityyppi on aina ilmoitettava. Dokumentin perusrakenne on siis

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  ...metainfoa...
</HEAD>
<BODY>
  ...dokumentin sisältö tähän...
</BODY>
</HTML>
```

Vaihtoehtoiset dokumenttityypin merkinnät ovat

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

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

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

HTML 4.01 Transitional -versiossa on jätetty mukaan joitakin ei-suositeltavia (*deprecated*) elementtejä ja attribuutteja, kuten **CENTER**, **FONT**, **STRIKE**, **U**, kuvien **BORDER**, **ALIGN**, **HSPACE** ja **VSPACE**, taulukkojen **BGCOLOR** ja **NOWRAP** sekä kaikki **BODY**-elementin attribuutit.

Dokumentin kieli voidaan ilmoittaa dokumentin alun **HTML**-tagissa **LANG**-attribuutilla. Englanti on **en**, amerikanenglanti on **en-US**, brittienglanti on **en-GB**, suomi on **fi** ja ranska on **fr**. Ruotsi on **sv** ja suomenruotsi on **sv-FI**. Muut koodit näet vaikka selaimesi kielisäädöistä. Esimerkki:

```
<HTML LANG="fi">
```

HEAD

Sisältää metainfoa joka ei näy sivun varsinaisessa sisällössä: selaimen kirjanmerkeissä ja otsikkorivillä näkyvä dokumentin nimi, tekijän nimi, tekijän sähköpostiosoite ja CSS-tyylisäännöt. Hakuroboteille voidaan kirjoittaa avainsanat ja lyhyt kuvaus dokumentista. Skriptit voidaan sijoittaa joko tänne tai **BODY**-osuuden sisälle. Käsittelen tarkemmin tänne sijoitettavia elementtejä luvussa [HEAD-elementit](#).

BODY

- ◆ `background=URL` (taustakuva) (vanhentunut)
- ◆ `bgcolor=väri` (taustaväri) (vanhentunut)
- ◆ `text=väri` (tekstin väri) (vanhentunut)
- ◆ `link=väri` (linkkien väri) (vanhentunut)
- ◆ `vlink=väri` (linkit joissa käyty) (vanhentunut)
- ◆ `alink=väri` (aktiivisen linkin väri) (vanhentunut)
- ◆ `onload=skripti` (tullessa sivulle)
- ◆ `onunload=skripti` (lähdettäessä sivulta)

Dokumentin sisältö tulee **BODY**-elementin sisälle. Kaikki yllä olevat **BODY**-elementin attribuutit lukuun ottamatta kahta viimeistä ovat *deprecated* (ei-suositeltavia) HTML 4 -suosituksessa. Niiden sijasta suositellaan käytettäväksi [CSS-tyylisääntöjä](#). Attribuuteilla voidaan ilmoittaa tekstin väri, taustaväri, mahdollinen taustakuva ja linkkien värit:

```
<BODY BACKGROUND="kuva.jpg" BGCOLOR="#FFFBF0" TEXT="black"
LINK="blue" VLINK="red" ALINK="lime">
```

Jos ilmoitat yhden näistä, ilmoita loputkin. Tällä varmistetaan se että väreissä ei ole lukemista haittaavia ristiriitoja. Jos taustakuvaa ei ladata tai sitä ei löydy palvelimelta, käytetään `BGCOLOR`-attribuutissa ilmoitettua taustaväriä. Saman esimerkin värit ilmoitettaisiin CSS:n avulla tällä tavalla:

```
<HEAD>
  <STYLE TYPE="text/css">
    a:link { color: blue }
    a:visited { color: red }
    a:active { color: lime }
    body { color: black;
           background-color: #FFFBF0;
           background-image: url(kuva.jpg) }
  </STYLE>
</HEAD>
```

HEAD-elementit

TITLE, META, STYLE, SCRIPT ja LINK

Nämä elementit ovat **HEAD**-osuuden sisällä. Ne sisältävät ”metainformaatiota”, joka ei näy varsinaisella sivulla. Voit ilmoittaa esimerkiksi sivun tekijän nimen ja sähköpostiosoitteen. Sivusta voidaan kirjoittaa lyhyt kuvaus. Ainoastaan **TITLE** on pakollinen. Esimerkki:

```
<HEAD>
<META NAME="Keywords" CONTENT="hakusana1, hakusana2, ..." >
<META NAME="Description" CONTENT="lyhyt kuvaus">
<META NAME="Author" CONTENT="tekijä">
<LINK REV="Made" HREF="mailto:tunnus@isp.fi">
<TITLE>Otsikko</TITLE>
  <STYLE TYPE="text/css">
    ...tänne CSS-tyylisäännöt...
  </STYLE>
  <SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript">
    ...tänne skripti...
  </SCRIPT>
</HEAD>
```

TITLE sisältää selaimen otsikkorivillä ja kirjanmerkeissä näkyvän dokumentin nimen. Nimen pitäisi vastata hyvin dokumentin sisältöä, ja sen pitäisi toimia missä tahansa asiayhteydessä. Jos joku lisää sivusi kirjanmerkkeihin, pelkkä ”Sisällys” on aika huono nimi. **LINK**- ja **META**-elementtien lopputagit **</LINK>** ja **</META>** ovat kiellettyjä. **LINK**-elementillä voidaan ilmoittaa tekijän sähköpostiosoite. **META**-elementeillä on esimerkissä ilmoitettu tekijän nimi sekä hakusanat ja dokumentin kuvaus hakukoneita varten.

META-elementeille ei ole olemassa standardoitua listaa; voit käyttää niitä niin paljon kuin haluat. HTML-koodi voi sisältää metainfoa myös sivulla käytetystä enkoodauksesta, kielestä, tyylikielestä ja skriptikielestä. Kaikki **META HTTP-EQUIV** -tagit on tarkoitettu palvelimen antaman informaation korvikkeiksi. Ne voidaan pottaa myös suoraan sivuun:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
<META HTTP-EQUIV="Content-Language" CONTENT="fi">
```

Sivun enkoodauksen ilmoittaminen kuuluu palvelimelle. Lisätietoja asiasta on [Merkistöistä](#)-sivulla. **META**-elementille on kuitenkin käyttöä, koska kaikilla ei ole mahdollisuutta säätää palvelimen asetuksia. Jos koodaus on ISO-8859-1, ongelmia ei yleensä tule. Jos käytetään jotain muuta koodausta, suosittelen **META**-elementin käyttöä.

Käytettävä skriptikieli pitää ilmoittaa vain, jos dokumentin HTML-elementteihin on liitetty ns. tapahtumamääritteitä (**onmouseover**, **onclick**, **onload**, **onunload** jne.). Tyylikieli ilmoitetaan **HEAD**-osuudessa vain, jos sivulla on käytetty **STYLE**-attribuuttia eli ”inline-tyylejä” HTML-elementtien yhteydessä.

```
<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">
<META HTTP-EQUIV="Content-Script-Type" CONTENT="text/javascript">
```

Jos sivusi siirtyvät toiselle palveluntarjoajalle ja haluat automaattisen uudelleenohjauksen, voit käyttää **META**-elementtiä, joka ohjaa selaimen uuteen osoitteeseen esim. 10 sekunnin kuluttua. Suositeltavampi tapa on hoitaa asia palvelimella.

```
<META HTTP-EQUIV="Refresh" CONTENT="10; URL=http://www.uusiosoite.fi/">
```

Jos haluat että hakurobotit eivät indeksoi sivua mutta seuraavat linkkejä toisille sivuille, käytä alla olevan esimerkin mukaista koodia. Oletusvaihtoehto on "all" (sama kuin "index, follow"), joka pyytää robotteja indeksoimaan sivun ja seuraamaan linkkejä.

```
<META NAME="Robots" CONTENT="noindex, follow">
```

CSS voidaan kirjoittaa suoraan dokumentin lähdekoodiin. Jos käytetään samaa ulkoista CSS-tyylisivua sivuston kaikille sivuille, siihen linkitetään seuraavalla tavalla:

```
<LINK REL="Stylesheet" TYPE="text/css" HREF="muntyyli.css">
```

Sivu voi käyttää useampia CSS-tyylisivuja, jolloin lukija voi vaihtaa esimerkiksi sivun väriteeman. Kaikki selaimet eivät tarjoa mahdollisuutta vaihtoehtoisten CSS-tyylitiedostojen valitsemiseen. Jos sellainen mahdollisuus tarjotaan, **TITLE**-attribuutilla ilmoitetun tekstin pitäisi näkyä selaimen säädöissä. Jos haluat käytännön esimerkin, kokeile [Web-julkaiseminen](#)-sivua.

```
<LINK REL="stylesheet" TYPE="text/css" HREF="white.css"
TITLE="Valkoinen (oletus)">
<LINK REL="alternate stylesheet" TYPE="text/css" HREF="black.css"
TITLE="Musta">
```

Sivulle voidaan tehdä oma CSS-tyylisivu printtaamista varten. **MEDIA**-attribuutin oletusarvo on **screen** eli tietokoneen näyttö. Muita mahdollisia arvoja ovat mm. **print** (tulostamiseen), **handheld** (kännyköille ja PDA-laitteille), **aural** (puhesyntetisaattoreille) ja **tv**. Printti-versiossa voit ehkä poistaa taustakuvan ja muuttaa sivun fonteja.

```
<LINK REL="Stylesheet" MEDIA="screen" HREF="selain.css" TYPE="text/css">
<LINK REL="Stylesheet" MEDIA="print" HREF="printti.css" TYPE="text/css">
```

LINK voi ilmoittaa myös metainfoa useamman sivun sisältävän sarjan edellisestä ja seuraavasta dokumentista. Internet Explorer ja useimmat muut graafiset selaimet eivät tue tätä HTML-standardien mukaista ominaisuutta. Tekstipohjainen Lynx tukee sitä.

```
<LINK REL="Prev" HREF="doc4.html" TITLE="Dokumentti 4">
<LINK REL="Up" HREF="." TITLE="Hakemisto">
<LINK REL="Next" HREF="doc6.html" TITLE="Dokumentti 6">
```

STYLE sisältää CSS-tyylisäännöt. Tässä esimerkissä `<P CLASS="box">`- ja `</P>`-tagien välissä oleva kappale (toivottavasti) renderoidaan Arial-fontilla, kappaleen taustaväri on vaaleansininen, kappale ympäröidään yhden pikselin mustalla viivalla, kappaleen sisällön ja viivan väliin jätetään 8 pikseliä tyhjää tilaa ja marginaalit ovat hieman suuremmat kuin sivun perusmarginaalit. CSS-koodi on tässä:

```
<HEAD>
  <STYLE TYPE="text/css">
    p.box { font-family: Arial,Helvetica,sans-serif;
            background-color: #F0F8FF;
            border: 1px solid black; padding: 8px;
            margin-left: 6%; margin-right: 10% }
  </STYLE>
</HEAD>
```

Attribuutti **TYPE** on pakollinen **STYLE**- ja **SCRIPT**-elementtien yhteydessä.

Skriptit voivat olla joko **BODY**- tai **HEAD**-elementin sisällä. Vanhentunut **LANGUAGE**-attribuutti on mukana HTML 4 Transitionalissa. Se on mukana yhteensopivuussyistä. Jotkut selaimet eivät aja skriptiä, jos sen tyyppiä (JavaScript, VBScript jne.) ei ole ilmoitettu **LANGUAGE**-attribuutilla:

```
<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript">
```

Generiset lohkoelementit

HI, H2, H3, H4, H5, H6

- ♦ **align**=[left|center|right|justify] (horisontaalinen kohdistus) (vanhentunut)

Dokumentin pääotsikko on **H1**, alaotsikot **H2...H6**. **H1**-tason otsikkoja ei saisi olla enempää kuin yksi. Kolmatta tai neljättä tasoa alemmaksi ei kannata mennä, dokumentin rakenne muuttuu liian epäselväksi. Selaimet renderoivat otsikkojen fonttikoot laskevassa järjestyksessä. Otsikkotasoja ei kuitenkaan pidä käyttää fonttien koon manipuloimiseen. HTML on tarkoitettu dokumentin rakenteen kuvaamiseen. CSS-tyylisäännöillä voidaan vaikuttaa sivun ulkoasuun. **H5** ja **H6** ovat sitä paitsi selaimissa usein *pienempiä* kuin normaali leipäteksti!

<H1 ALIGN="center"> voidaan korvata monipuolisemmalla CSS-tyylillä. Esimerkissä **H1** on sivun keskellä, fontin koko on kaksi kertaa suurempi kuin leipätekstin koko, otsikon alapuolelle tulee 35 pikseliä tyhjää tilaa ja otsikon väri on vaaleansävyinen violetti:

```
<HEAD>
  <STYLE TYPE="text/css">
    h1 { text-align: center;
        font-size: 2em;
        margin-bottom: 35px;
        color: #DB7093 }
  </STYLE>
</HEAD>
```

Voit tehdä jokaisesta otsikosta linkin kohteen:

```
<H2><A NAME="toc">Sisällys</A></H2>
```

Tämän jälkeen otsikkoon voidaan linkittää normaalisti dokumentin muista osista:

**** on linkki **toc**-nimellä identifioituun otsikkoon.

P

- ◆ `align=[left|center|right|justify]` (horisontaalinen kohdistus) (vanhentunut)

P on normaali tekstikappale. Se ei voi pitää sisällään muita lohkoelementtejä (esim. **DIV**, **BLOCKQUOTE**, **TABLE**, **PRE**, **UL**). Lopputagi `</P>` ei ole pakollinen. Sitä kannattaa kuitenkin käyttää selainten bugien takia. Tyhjää **P**-kappaletta ei saisi käyttää ylimääräisen tyhjän tilan luomiseen. Attribuutti **ALIGN** määrittää tekstin tasauksen: **left** (oletus) tasaa vasempaan reunaan, **right** oikeaan, **center** keskittää tekstin ja **justify** tasaa molemmat reunat. Molempien reunojen tasaus ei kaikissa selaimissa toimi kunnolla.

Jos fonttia ei erikseen määritellä, se on Windowsissa yleensä 12 pisteen Times New Roman, riippuen tietysti käyttäjän asetuksista. Kappaleiden väliin selaimet pistävät yhden tyhjän rivin. Kappaleiden ensimmäisiä rivejä ei sisennetä. Seuraavassa CSS-esimerkissä suositeltu fontti on 16 pikselin Verdana, kappaleiden ensimmäiset rivit sisennetään ja kappaleiden välissä ei ole tyhjää tilaa:

```
p { font-family: Verdana,Arial,Helvetica,sans-serif;
    font-size: 16px;
    text-indent: 2em;
    margin-top: 0;
    margin-bottom: 0 }
```

CSS:ssä voidaan käyttää fonteille suhteellisia kokomäärittelyjä: **em**, **smaller**, **larger** tai prosentit. Vältä absoluuttisia kokoja, joihin kuuluvat mm. pikselit (**px**) ja pisteet (**pt**). Jos jätät **P**-elementtien sisällä olevan fontin koon ilmoittamatta, sivut skaalautuvat paljon paremmin eri tilanteisiin. (Lisää fonteista sivulla [Web-julkaiseminen](#).)

DIV

- ◆ `align=[left|center|right|justify]` (horisontaalinen kohdistus) (vanhentunut)

DIV on geneerinen lohkoelementti, jota käytetään esimerkiksi dokumentin tietyn osan fontin tai taustaväriin tai sijoittelun apuna. Se on semanttisesti tyhjä elementti. Apuna on CSS jolla määritellään kunkin **DIV**-osan tyyli. Kaksipalstaiset layoutit tehdään usein **DIV**-elementtien avulla. **DIV** voi pitää sisällään mitä tahansa lohko- ja tekstielementtejä, myös toisen **DIV**-elementin. Vanhentunut `<DIV ALIGN="center">` on sama asia kuin `<CENTER>`. Tässä on yksinkertainen CSS-esimerkki:

```
#foo { float: left;
       width: 160px }
#bar { background-color: #F5F5F5;
       margin-left: 10px }
```

Tämän jälkeen `<DIV ID="foo">`- ja `</DIV>`-tagien välissä oleva osa kelluu vasemmalla. Osan leveys on 160 pikseliä. `<DIV ID="bar">` aloittaa osan, jonka taustaväri on vaaleanharmaa ja vasen marginaali 10 pikseliä.

BLOCKQUOTE

- ◆ `cite=URL` (lainauksen lähde)

Lohkositaatti **BLOCKQUOTE** sisältää lainauksen ulkopuolisesta lähteestä. Toisin kuin inline-lainaus **Q**, se on lohkotason elementti, joka voi pitää sisällään muita lohkoelementtejä (**P**, **TABLE** jne.). **BLOCKQUOTE** ei voi olla **P**-elementin tai tekstielementtien sisällä. Selaimet esittävät **BLOCKQUOTE**-lainauksen yleensä sisennettynä, ja jos olet lukenut tähän saakka, varmaan arvaat että tätä elementtiä ei suositella käytettäväksi pelkän sisennyksen aikaansaamiseen. Esimerkissä lainauksen teksti on HTML 4 Strict -suosituksen mukaisesti sijoitettu **P**-tekstikappaleen sisälle:

```
<P>Ote Saarikosken suomentamasta <CITE>Kravun kääntöpiiristä</CITE>:</P>
<BLOCKQUOTE>
<P>Minulla ei ole rahaa, ei resursseja, ei toiveita. Minä olen
maailman onnellisin ihminen. Vuosi sitten, puoli vuotta sitten,
minä ajattelin olevani taiteilija. Enää minä en ajattele, minä
<I>olen</I>. Kaikki mikä oli kirjallisuutta on karissut minusta.
Enää ei ole kirjoja jotka olisi syytä kirjoittaa, Jumalan kiitos.</P>
</BLOCKQUOTE>
```

PRE

- ◆ `width=numero` (rivin pituus) (vanhentunut)

Tätä käytetään valmiiksi muotoillun tekstin esittämiseen. Selaimet esittävät **PRE**-elementin sisällä olevan tekstin kiinteävälisellä fontilla (esim. Courier). Välilyönnit, tabulaattorit ja rivinvaihdot säilytetään alkuperäisen tekstin mukaisina. **PRE** on hyödyllinen esimerkiksi silloin, kun halutaan esittää ohjelmakoodia tai HTML-koodia (tämänkin sivun kaikki esimerkit ovat **PRE**-elementtien sisällä). **PRE** ei voi olla normaalin **P**-kappaleen tai **DT**-elementin sisällä, mutta voi olla minkä tahansa muun lohkoelementin sisällä. **PRE** voi pitää sisällään mitä tahansa tekstielementtejä lukuun ottamatta seuraavia: **IMG**, **OBJECT**, **APPLET**, **BIG**, **SMALL**, **SUB**, **SUP** ja **FONT**.

Vanhentunut **WIDTH**-attribuutti kertoo selaimelle kuinka pitkiä rivien pitäisi olla. Selaimet eivät kuitenkaan ikinä tukeneet tätä attribuuttia. Esimerkki C-koodin esittämisestä:

```
<PRE>
printk(KERN_EMERG "Kernel panic: %s\n",buf);
if (current == task[0])
    printk(KERN_EMERG "In swapper task - not syncing\n");
else if (in_interrupt())
    printk(KERN_EMERG "In interrupt handler - not syncing\n");
else
    sys_sync();

unblank_console();
</PRE>
```

Jos esität koodia **PRE**-elementin sisällä, muuta kaikki tabulaattorit välilyönneiksi—et voi tietää käyttäkö lukijan selain kahdeksan vai neljän lyönnin tabulaattoria, ja pahimmassa tapauksessa muotoilu menee täysin sekaisin. Merkit `<`, `>` ja `&` pitää muuttaa [symbolisiksi notaatioiksi](#) `<`, `>` ja `&`.

HR

- ◆ `align=[left|center|right]` (horisontaalinen kohdistus) (vanhentunut)
- ◆ `width=[pikselit|prosentit]` (viivan leveys) (vanhentunut)
- ◆ `size=pikselit` (viivan korkeus) (vanhentunut)
- ◆ `noshade` (ei varjostusta) (vanhentunut)

Horisontaalinen viiva, joka jakaa dokumentin eri osia. **HR**-viivalla voi erottaa esimerkiksi linkkilistan tai osoitetiedot dokumentin muusta sisällöstä. Lopputagi `</HR>` on kielletty. Viivan leveyden (`WIDTH`) voi ilmoittaa pikseleinä tai prosentteina. Käytä mieluummin prosentteja (esim. `<HR WIDTH="80%">`), viiva skaalautuu silloin paremmin eri resoluutioille. CSS:n avulla voi muokata viivan ulkoasua. Voit kirjoittaa esimerkiksi

```
hr { color: #708090;
      height: 4px }
```

jonka pitäisi saada aikaan tummanharmaa viiva, jonka paksuus on neljä pikseliä.

ADDRESS

on lohkoelementti joka pitää sisällään dokumentin tekijän nimen, sähköpostiosoitteen, postiosoitteen, puhelinnumeron, linkin tekijän kotisivulle ja muuta vastaavaa. Yleensä yhteystiedot sijoitetaan sivun loppuun. **ADDRESS** ei voi olla normaalin **P**-kappaleen tai **DT**-elementin sisällä, mutta voi olla minkä tahansa muun lohkoelementin sisällä. HTML 4 Transitional sallii **P**-elementin **ADDRESS**-elementin sisälle, HTML 4 Strict sallii ainoastaan tekstielementit. Esimerkki:

```
<ADDRESS>Markku Immonen
<A HREF="mailto:wired.poet@sci.fi">wired.poet@sci.fi</A>
</ADDRESS>
```

Selaimet näyttävät sisällön yleensä kursivoituna:

Markku Immonen <wired.poet@sci.fi>

NOSCRIPT (HTML 4.0)

Uusi HTML 4 -suositukseen kuuluva **NOSCRIPT** tarjoaa vaihtoehdoisen sisällön, jos sivulla olevan skriptin ajaminen ei jostain syystä onnistu tai jos käyttäjä on disabloinut skriptit. Sen pitäisi tulla heti korvattavan **SCRIPT**-elementin jälkeen. Tästä ei ole paljon hyötyä—jos sivulla on skripti, se on yleensä olennainen osa sivua eikä sen korvaaminen ole helppoa staattisella sisällöllä. Jotkut selaimet eivät näytä **NOSCRIPT**-elementin sisältöä, jos skriptikieli ei ole tuettuna selaimessa. Sisältö näytetään ehkä vain silloin kun skriptit on käyttäjän toimesta disabloitu.

CENTER (vanhentunut)

Keskittää dokumentin tietyn osan sisällön. `<DIV ALIGN="center">` toimii täsmälleen samalla tavalla. Jos käytät **CENTER**-elementtiä taulukon keskittämiseen, vain taulukko keskitetään, ei yksittäisten taulukkosolujen sisältöä. Esimerkki:

```
<CENTER>
<TABLE>
  <TR ALIGN="center">
    <TD>Solu 1</TD>
    <TD>Solu 2</TD>
  </TR>
</TABLE>
</CENTER>
```

Tekstielementit

A

- ◆ **href=URL** (linkin kohteen URL-osoite)
- ◆ **name=merkkijono** (sivun sisäisissä linkeissä jos A on linkin kohteena)
- ◆ **type=mediatyyppi** (linkin kohteen mediatyyppi)
- ◆ **hreflang=kielikoodi** (linkin kohteen kieli)
- ◆ **charset=enkoodaus** (linkin kohteen enkoodaus)
- ◆ **target=kehys** (missä kehyksessä linkin kohde renderoidaan) (vanhentunut)

A on hypertekstilinkki ("ankkuri" hypertekstilinkille). Attribuutti **href** ilmoittaa linkin kohteen osoitteen. Lisätietoina voidaan ilmoittaa linkin kohteen enkoodaus ja mediatyyppi (MIME-tyyppi). Ensimmäisessä esimerkissä linkitetään AU-äänitiedostoon ja toisessa sivuun jonka enkoodauksena on ISO-8859-2:

```
Jos haluat kuulla miten Linus Torvalds lausuu sanan "Linux",
kuuntele tämä <A HREF="linus.au" TYPE="audio/basic">äänitiedosto</A>.
Prahaan matkustaville suosittelen <A HREF="czh.html"
CHARSET="ISO-8859-2">tshekin kielioppia</A>.
```

Yleisiä Internet-mediatyyppejä ovat esimerkiksi

- ◆ audio/wav (WAV), audio/basic (AU, SND), audio/midi (MID, MIDI), audio/mpeg (MP3), audio/x-ms-wma (WMA)
- ◆ video/mpeg (MPG, MPEG), video/quicktime (MOV, QT), video/x-msvideo (AVI), video/x-ms-asf (ASF)
- ◆ image/jpeg (JPG, JPEG), image/gif (GIF), image/png (PNG), image/tiff (TIF, TIFF)
- ◆ application/pdf (PDF), application/zip (ZIP), application/octet-stream (EXE, BIN, CLASS jne. = luokittelematon binääridata), application/java tai application/x-java-applet (CLASS), application/msword (DOC), application/vnd.ms-excel (XLS)
- ◆ text/plain, text/html, application/xhtml+xml, text/xml, text/css, text/javascript

Linkin yhteydessä voidaan ilmoittaa myös linkin kohteen kieli:

```
Lue selostus <A HREF=" ../dernier.html" HREFLANG="fr">Charles
Baudelairin viimeisistä päivistä</A>.
```

Englanti on **en**, amerikanenglanti on **en-US**, brittienglanti on **en-GB**, suomi on **fi** ja ranska on **fr**. Ruotsi on **sv** ja suomenruotsi on **sv-FI**. Muut koodit näet vaikka selaimesi kielisäädöistä.

NAME-attribuuttia käytetään sivun sisäisissä linkeissä. Jos haluat tehdä linkin saman sivun toiseen sektioon, merkitse linkin kohde tällä tavalla:

```
<H2><A NAME="foobar">Otsikko</A></H2>
```

Nyt esimerkin **H2**-otsikko voi olla linkin kohteena. Ja kun teet tähän otsikkoon osoittavan linkin, osoite on **HREF="#foobar"**. Jos linkitetään toiselta sivulta, osoite on **HREF="sivu.html#foobar"**. HTML 4 -suosituksen [yhteisiin attribuutteihin](#) kuuluva **ID** on suunniteltu korvaamaan **NAME**. Jos käytät linkin kohteessa sekä **NAME**- että **ID**-attribuutteja, varmista että molempien arvo on sama.

Yhteisiin attribuutteihin kuuluva **TITLE** on hyödyllinen **A**-linkkien yhteydessä. **TITLE**-attribuutin arvona voi olla lyhyt kuvaus linkin kohteesta, jolloin selain voi näyttää ”tooltip”-tekstin hiiren kursorin ollessa linkin kohdalla.

```
<A HREF="http://www.nytimes.com/"
TITLE="New York Timesin sivuilla pitää rekisteröityä">
New York Times</A>
```

Selaimesi näyttää esimerkin näin:

[New York Times](#)

Vanhentunut **TARGET** kertoo selaimelle missä kehyksessä (frame) linkin kohde renderoidaan. Kehyksen nimen sijasta voidaan käyttää näitä arvoja:

- ◆ **_blank** (näyttää linkin kohteen uudessa nimeämättömässä ikkunassa; toimii myös sivuilla jotka eivät käytä kehyksiä)
- ◆ **_self** (näyttää linkin kohteen senhetkisessä kehyksessä)
- ◆ **_parent** (näyttää linkin kohteen saman sivun toisessa kehyksessä)
- ◆ **_top** (näyttää linkin kohteen samassa kokonaisessa ikkunassa ilman kehyksiä)

BR

- ◆ **clear=[left|all|right|none]** (tee tyhjää tilaa) (vanhentunut)

Pakottaa rivinvaihdon. Lopputagi **</BR>** on kielletty. Käytä **BR**-tagia, jos rivinvaihtojen kohdat on taiteellisen vaikutelman takia pakko ilmoittaa. Tyypillisiä esimerkkejä ovat osoitetiedot ja **CODE**-elementin sisällä oleva ohjelmakoodi. Älä käytä **
-tagia normaaleissa tekstikappaleissa—tuloksena on useimmiten sotkua, jos sivuasi katsotaan erikokoisella fontilla tai resoluutiolla. **CLEAR lisää tyhjää rivejä rivinvaihdon perään silloin kun tekstikappale sisältää ”kelluvan” kuvan. Nyt on ASCII-taiteen aika:

```

***** -----
|      | -----
| kuva | ---<BR CLEAR="left">
|      | -----
*****
-----
-----

***** -----
|      | -----
| kuva | ---<BR>
|      | -----
*****
-----
-----

```

Käytä attribuuttia `CLEAR="all"`, jos kuvia on molemmissa reunoissa. CSS auttaa myös rivinvaihdossa:

```
br.vasenkuva { clear: left }
```

Tyylisäännön jälkeen `<BR CLASS="vasenkuva">`-tagi tekee tyhjää tilaa tekstikappaleen vasemmalle puolelle.

IMG

- ◆ `src=URL` (kuvan lähde)
- ◆ `alt=merkkijono` (vaihtoehtoinen teksti)
- ◆ `width=pikselit` (kuvan leveys)
- ◆ `height=pikselit` (kuvan korkeus)
- ◆ `usemap=URL` (käytettävän kuvakartan osoite)
- ◆ `name=merkkijono` (identifioi kuvan esim. JavaScriptille)
- ◆ `align=[top|middle|bottom|left|right]` (kuvan kohdistus) (vanhentunut)
- ◆ `border=pikselit` (kuvan reunuksen leveys) (vanhentunut)
- ◆ `hspace=pikselit` (tila kuvan vas. ja oik. puolella) (vanhentunut)
- ◆ `vspace=pikselit` (tila kuvan ylä- ja alapuolella) (vanhentunut)

Kuvan lopputagi `` on kielletty. HTML 4 -suosituksessa `IMG` on aina lohkoelementin sisällä (esim. `P`, `TD`, `LI`). Pakollinen attribuutti `SRC` kertoo mistä kuva löydetään. `ALT` sisältää vaihtoehtoisen tekstin lukijoille, jotka käyttävät tekstipohjaista selainta tai ovat ottaneet selaimesta kuvien latauksen pois päältä. Teksti kannattaa harkita tarkkaan: sen ei pitäisi täydentää kuvaa vaan *korvata* se. HTML 4:ssa `ALT` on pakollinen.

Kuvan korkeus ja leveys kannattaa ilmoittaa totuudenmukaisesti, koska selaimet eivät yleensä osaa kunnolla skaalata kuvaa uuteen kokoon. Kuvat ovat selainten rajoituksista johtuen melkein aina GIF-, JPG- tai PNG-formaatissa. Muut formaatit näytetään usein pluginien avulla. ([Väreistä](#)-sivulla on enemmän juttua kuvaformaateista.) Yksinkertainen esimerkki:

```
<IMG SRC="images/kissa.jpg" WIDTH="120" HEIGHT="80" ALT="[Garfield]">
```

`ALIGN`-attribuuttia voidaan käyttää silloin, kun kuva on tekstikappaleen sisällä. Arvot `top`, `middle` ja `bottom` määrittävät mihin kohtaan tekstiriviä kuva sijoittuu vertikaalisesti. Arvot `left` ja `right` saavat aikaan ”kelluvan” kuvan (ks. yllä `BR`-elementin kuvaus). Jos kuva toimii linkkinä, selaimet piirtävät sen ympärille sinisen reunuksen. `BORDER="0"` eliminoi reunuksen. `HSPACE` ja `VSPACE` määrittävät kuvan ympärille jätettävän tyhjän tilan pikseleinä. Kuten aina, CSS on tässäkin tapauksessa suositeltavampi vaihtoehto. Esimerkissä kuva kelluu oikealla, sen vasemmalla puolella ja alapuolella on 10 pikseliä tyhjää tilaa ja reunuksia ei laiteta kuvan ympärille:

```
img#kissa { float: right;
padding-left: 10px;
padding-bottom: 10px;
border: 0 }
```

Vanhojen selainten CSS-tuki oli aika puutteellinen, varsinkin yllä olevan esimerkin kaltaisissa jutuissa. HTML:ssä sama lopputulos saataisiin aikaan näin:

```
<IMG SRC="images/kissa.jpg" WIDTH="120" HEIGHT="80" ALT="[Garfield]"
ALIGN="right" HSPACE="10" VSPACE="10" BORDER="0">
```

Attribuuttia **USEMAP** käytetään silloin kun kuva toimii kuvakarttana (image map). Attribuutin arvo viittaa samalla sivulla olevaan **MAP**-elementtiin. Jos **MAP**-elementille on annettu nimi **nav**, varsinaisen kuvan yhteydessä siihen viitataan linkillä ``.

MAP

- ◆ **name=merkkijono** (kuvakartan nimi)

MAP on kuvakartta (image map). Se voi olla lohkoelementti tai tekstielementti. Sisältönä on yksi tai useampia **AREA**-elementtejä. Pakollinen attribuutti **NAME** toimii ankkurina **IMG**- tai **OBJECT**-elementin attribuutille **USEMAP**. Voit antaa kartalle minkä tahansa nimen.

```
<IMG SRC="kuva.jpg" USEMAP="#nav" ALT="[Navigointikartta]">
<MAP NAME="nav">
...image map tähän...
</MAP>
```

Kaikki selaimet eivät tue **OBJECT**-elementtiä, joten perinteisten **IMG**-kuvien käyttäminen tässä yhteydessä voi olla hyvä idea. Kuvakartan lisäksi kannattaa tarjota tekstilinkeillä toteutettu vaihtoehtoinen navigointiratkaisu. Tarkempi selostus kuvakartoista on **AREA**-elementin yhteydessä.

AREA

- ◆ **shape=[rect|poly|circle|default]** (alueen muoto)
- ◆ **coords=pikselit** (alueen koordinaatit)
- ◆ **href=URL** (minne alue viittaa)
- ◆ **nohref** (alue ei toimi linkkinä)
- ◆ **alt=merkkijono** (vaihtoehtoinen teksti)
- ◆ **target=kehys** (missä kehyksessä linkin kohde renderoidaan) (vanhentunut)

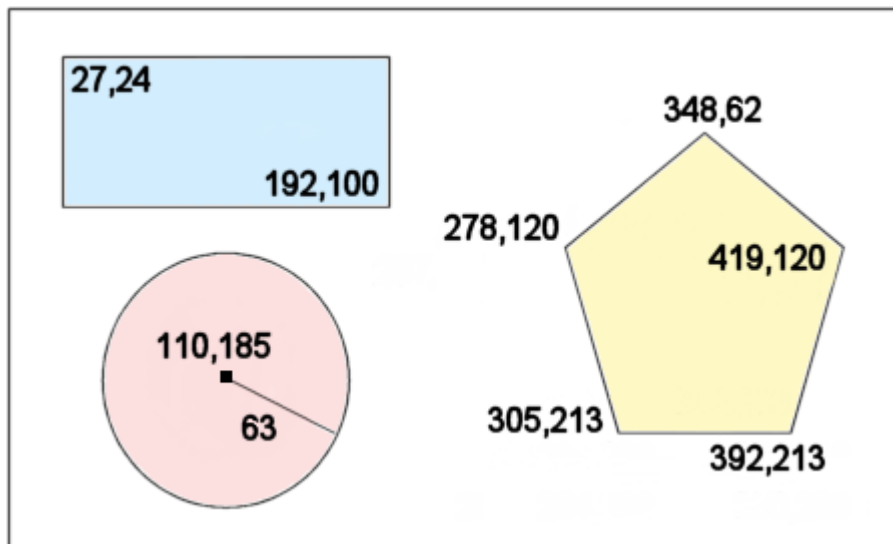
Nyt päästään käsittelemään kuvakarttoja vähän syvemmillä. **AREA** määrittää kuvakarttana toimivasta kuvasta tietyn linkkinä toimivan alueen. Lopputagi `</AREA>` on kielletty. Alueen sijainti kuvassa määritellään **SHAPE**- ja **COORDS**-attribuuteilla. Alue voi olla suorakulmio, polygoni tai ympyrä. **HREF** viittaa linkin kohteeseen, ja pakollinen **ALT** sisältää vaihtoehtoisen tekstin aivan kuten normaaleissa **IMG**-kuvissakin. Esimerkki:

```

<MAP NAME="nav">
<AREA SHAPE="rect" COORDS="27,24,192,100"
  HREF="sivu1.html" ALT="Sivu 1">
<AREA SHAPE="poly" COORDS="348,62,419,120,392,213,305,213,278,120,348,62"
  HREF="sivu2.html" ALT="Sivu 2">
<AREA SHAPE="circle" COORDS="110,185,63"
  HREF="sivu3.html" ALT="Sivu 3">
<AREA SHAPE="default" NOHREF ALT="">
</MAP>

```

Suorakulmion (**rect**) tapauksessa koordinaattien ensimmäinen luku kertoo horisontaalisen etäisyyden kuvan vasemmasta reunasta ja toinen luku tarkoittaa vertikaalista etäisyyttä kuvan yläreunasta. Nämä kaksi lukua (27,24) määrittävät linkkialueen *vasemman yläkulman* paikan. Seuraava lukupari (192,100) tarkoittaa linkkialueen *oikeaa alakulmaa*. Polygonissa (**poly**) koordinaatit ilmoitetaan järjestyksessä alkaen mistä tahansa kulmasta, ja lopuksi kuvio suljetaan toistamalla ensimmäisen pisteen koordinaatit. Ympyrässä (**circle**) ensimmäinen lukupari (110,185) tarkoittaa ympyrän keskipistettä ja viimeinen luku (63) ympyrän sädettä pikseleinä.



Mitä tehdään alueella joka ei toimi linkkinä? Ei mitään. **SHAPE="default"** tarkoittaa ylimääräistä aluetta. Esimerkissä sille on annettu attribuutti **NOHREF**, jolloin alue ei toimi linkkinä. Toimiville linkkialueille on annettu normaali **ALT**-teksti, ja **default**-alueen vaihtoehtoisena tekstinä on pelkkää tyhjää (**ALT=""**).

SCRIPT

- ◆ **type=mediatyyppi** (mediatyyppi)
- ◆ **src=URL** (ulkopuolisen skriptin osoite)
- ◆ **charset=enkoodaus** (ulkopuolisen skriptin enkoodaus)
- ◆ **language=merkkijono** (skriptikieli) (**vanhentunut**)

Skriptejä voi esiintyä yhden tai useamman kerran joko **HEAD**- tai **BODY**-elementin sisällä. **SCRIPT** voi toimia joko lohko- tai tekstielementtinä. Pakollinen **TYPE** kertoo skriptin tyyppin. Yleisin vaihtoehto on **TYPE="text/javascript"**. Jotkut selaimet vaativat vanhentuneen

LANGUAGE-attribuutin. Sitä suositellaan käytettäväksi **TYPE**-attribuutin lisäksi. Mahdollisia arvoja ovat esim. JavaScript, JavaScript1.2 ja VBScript. Pelkkä **<SCRIPT>**-tagi ei mene läpi validaattoreista.

SRC ilmoittaa ulkopuolisen skriptin osoitteen ja **CHARSET** ulkopuolisen skriptin enkoodauksen (melkein aina ISO-8859-1). Esimerkissä sivulla oleva skripti ajetaan vain jos ulkopuolista skriptiä ei pystytä ajamaan:

```
<SCRIPT SRC="munskripti.js" TYPE="text/javascript" LANGUAGE="JavaScript">
<!--
...vaihtoehtoinen skripti...
//-->
</SCRIPT>
```

Esimerkissä olevat kommenttimerkit varmistavat sen, että vanhat selaimet jättävät skriptin huomioimatta. Vanhemmille selaimille voi tarjota vaihtoehtoisen sisällön **NOSCRIPT**-elementillä.

Jos dokumentin HTML-elementteihin on liitetty ns. tapahtumamäärittäjiä (**onmouseover**, **onclick**, **onload**, **onunload** jne.), **HEAD**-elementin sisällä pitää ilmoittaa dokumentin oletusskriptikieli:

```
<HEAD>
<META HTTP-EQUIV="Content-Script-Type" CONTENT="text/javascript">
</HEAD>
```

SPAN (HTML 4.0)

SPAN on geneerinen tekstitason elementti, joka on semanttisesti tyhjä. (**DIV** on vastaava lohkotason elementti.) Sen yhteydessä voidaan käyttää vaikka [yhteisiin attribuutteihin](#) kuuluvaa **LANG**-attribuuttia ilmoittamaan **SPAN**-elementin sisällä olevan tekstin kieli. Yleensä **SPAN** esiintyy kuitenkin CSS-tyylisääntöjen apuna, jolloin siihen liitetään attribuutti **CLASS** tai **ID**. Esimerkissä **** aloittaa keltaisella taustavärillä esitetyn alleviivatun tekstin:

```
<HEAD>
  <STYLE TYPE="text/css">
    span.varoitus { background-color: #FFFFCC;
                   text-decoration: underline }
  </STYLE>
</HEAD>

...<SPAN CLASS="varoitus">tämä teksti on alleviivattuna
keltaisella taustalla</SPAN>...
```

Q (HTML 4.0)

- ◆ `cite=URL` (lainauksen lähde)

Q on tarkoitettu lyhyisiin inline-lainauksiin. Pitempiin lainauksiin sopii paremmin **BLOCKQUOTE**. Selaimen pitäisi laittaa automaattisesti lainausmerkit `<Q>`- ja `</Q>`-tagien kohdalle. Attribuuttia `LANG` voidaan käyttää tekstin kielen ilmoittamiseen esimerkiksi **P**- tai **SPAN**-elementtien yhteydessä. Lainausmerkkisäännöt vaihtelevat eri kielissä, ja selain voi päätellä `LANG`-attribuutin arvon perusteella käytetäänkö yksinkertaisia vai kaksinkertaisia lainausmerkkejä ja miten toimitaan sisäkkäisissä lainauksissa. Suomessa käytetään symmetrisiä kokolainausmerkkejä. Epäsymmetrisiä kokolainausmerkkejä, joissa lainauksen alun ja lopun merkit ovat erilaisia, käytetään mm. amerikanenglannissa. Kaksi esimerkkiä:

```
<P LANG="fi"><Q>En valehdellut</Q>, väitti pääministeri.</P>
<P LANG="en-US"><Q>Don't leave me,</Q> she said.</P>
```

Selaimen pitäisi esittää lainaukset näin:

”En valehdellut”, väitti pääministeri.
 “Don’t leave me,” she said.

Koska läheskään kaikki selaimet eivät tue **Q**-elementtiä, on paras unohtaa **Q** ja huolehtia itse lainausmerkeistä. Kaarevat kokolainausmerkit saadaan mukaan numeerisilla viittauksilla `“` (“ ”) ja `”` (” ’ ”). Puolilainausmerkkien positiot ovat `‘` (‘ ’) ja `’` (’ ’).

OBJECT (HTML 4.0)

- ◆ `data=URL` (objektin URL-osoite)
- ◆ `classid=tiedosto` (objektin implementaatio, esim. Java-sovelman luokkatiedosto)
- ◆ `codebase=URL` (polku objektin (data tai classid) hakemistoon)
- ◆ `type=mediatyyppi` (objektin (data) mediatyyppi)
- ◆ `codetype=mediatyyppi` (objektin (classid) mediatyyppi)
- ◆ `usemap=URL` (käytettävän kuvakartan osoite)
- ◆ `standby=merkkijono` (näytetään kun odotetaan objektin latautumista)
- ◆ `width=pikselit` (objektin leveys)
- ◆ `height=pikselit` (objektin korkeus)
- ◆ `align=[top|middle|bottom|left|right]` (objektin kohdistus) (vanhentunut)
- ◆ `border=pikselit` (objektin reunuksen leveys) (vanhentunut)
- ◆ `hspace=pikselit` (tila objektin vas. ja oik. puolella) (vanhentunut)
- ◆ `vspace=pikselit` (tila objektin ylä- ja alapuolella) (vanhentunut)

Uusi **OBJECT**-elementti voi sisältää esimerkiksi kuvan, kuvakartan, videon tai Java-sovelman. Kaikki selaimet eivät vielä tue tätä elementtiä. Se on tarkoitettu korvaamaan **IMG**- ja **APPLET**-elementit, Netscapen käyttämä **EMBED** ja Internet Explorerin **BGSOUND**. **OBJECT** voi olla joko lohko- tai tekstielementtinä. Seuraavassa yksinkertaisessa esimerkissä **OBJECT** viittaa kuvaan:


```
<OBJECT DATA="images/kissa.jpg" TYPE="image/jpeg"
WIDTH="120" HEIGHT="80">[Garfield]</OBJECT>
```

Huomaa että normaalissa **IMG**-kuvassa vaihtoehtoinen teksti esitetään **ALT**-attribuutilla, mutta **OBJECT**-elementin yhteydessä elementin *sisältö* näytetään silloin kun selain ei lataa kuvia. Sisältö näytetään myös silloin kun selain ei tue **OBJECT**-elementtiä. Seuraavassa esimerkissä **USEMAP**-attribuutti ilmoittaa kuvaan liittyvän kuvakartan (image map) sijainnin:

```
<OBJECT DATA="kuva.jpg" TYPE="image/jpeg" USEMAP="#nav"
WIDTH="300" HEIGHT="400"></OBJECT>
<MAP NAME="nav">
...image map tähän...
</MAP>
```

Seuraavassa esimerkissä **OBJECT** viittaa Java-sovelmaan. **CODEBASE** kertoo luokkatiedoston hakemiston polun, jos tiedosto ei ole samassa hakemistossa kuin HTML-dokumentti. **CLASSID**-attribuutilla ei siis ilmoiteta hakemistoa, pelkkä tiedoston nimi. **CODEBASE**-attribuutilla ilmoitetun hakemiston pitää olla alihakemisto. Hakemistopuussa ylöspäin menevät linkit (**CODEBASE**=".. /applets/") ovat turvallisuussyistä kiellettyjä.

```
<OBJECT CLASSID="java:my.class" CODEBASE="applets/"
CODETYPE="application/java" WIDTH="300" HEIGHT="400">
<P>Java-pluginia ei ole asennettu tai Java on kytketty pois
päältä selaimen asetuksista.</P>
</OBJECT>
```

OBJECT on joustavampi kuin **IMG** tai **APPLET**. Se antaa mahdollisuuden vaihtoehtoisten sisältöjen esittämiseen, jos selain ei tue ensisijaista objektia. **OBJECT**-elementin sisältö on korvaava vaihtoehto **<OBJECT>**-tagin sisällä ilmoitetulle objektille. Seuraavassa esimerkissä **OBJECT** viittaa Java-sovelmaan; jos selain ei pysty ajamaan sitä, näytetään QuickTime-video; jos sekään ei onnistu, näytetään animoitu gif-kuva; jos käyttäjä on ottanut kuvien latauksen pois päältä, näytetään tekstiviesti:

```
<OBJECT CLASSID="java:my.class" CODETYPE="application/java"
STANDBY="Odota..." WIDTH="300" HEIGHT="400">
  <OBJECT DATA="myvideo.mov" TYPE="video/quicktime"
  WIDTH="300" HEIGHT="400">
    <OBJECT DATA="kuva.gif" TYPE="image/gif" WIDTH="300" HEIGHT="400">
      <P>Saat datan osoitteesta <A HREF="foo.html">foo.html</A></P>
    </OBJECT>
  </OBJECT>
</OBJECT>
```

STANDBY-attribuutin teksti näytetään kun odotetaan Java-sovelman latautumista. Selaimet vaativat usein **WIDTH**- ja **HEIGHT**-määrittelyt upotetulle objektille. Olen tehnyt [testisivun](#), jossa voit testata tukeeko selaimesi **OBJECT**-elementtiä.

Internet Explorer ei ehkä aja Java-sovelmaa, jos **OBJECT**-elementtiä on käytetty. Vanhentunut **APPLET** toimii kaikissa selaimissa. Jos haluat käyttää HTML 4.01 Strict -dokumenttityyppiä, **OBJECT** on ainoa vaihtoehto. Ongelma ratkaistaan sisäkkäisillä **OBJECT**-elementeillä. Lisätietoja on Shayne Steelen sivulla [Java applet using XHTML object tag](#). IE käyttää esimerkkikoodissa sisempää objektia ja muut selaimet ulompaa:

```

<!--[if !IE]>-->
<OBJECT CLASSID="java:my.class" CODETYPE="application/java"
WIDTH="300" HEIGHT="400">
<!--<![endif]-->
  <OBJECT CLASSID="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
  WIDTH="300" HEIGHT="400">
    <PARAM NAME="code" VALUE="my.class">
      <P>Java-pluginia ei ole asennettu.</P>
    </OBJECT>
  <!--[if !IE]>-->
</OBJECT>
<!--<![endif]-->

```

PARAM (HTML 4.0)

- ◆ **name=merkkijono** (parametrin nimi)
- ◆ **value=merkkijono** (parametrin arvo)
- ◆ **valuetype=[data|ref|object]** (arvon tyyppi; oletus on data eli merkkijono)
- ◆ **type=mediatyyppi** (arvon kohteen mediatyyppi)

Upotetut objektit vaativat usein tiettyjä parametreja toimiakseen. **PARAM**-elementtiä käytetään tähän tarkoitukseen sekä **OBJECT**-elementin että vanhemman **APPLET**-elementin yhteydessä. Lopputagi **</PARAM>** on kielletty. Parametrit riippuvat tietysti upotetusta objektista tai käytetystä pluginista. QuickTime-videot hyväksyvät **autoplay**-parametrin, kun taas wav-äänien kohdalla vastaavan toiminnon nimi saattaa olla **autostart**. Esimerkissä korvataan Internet Explorerin **BGSOUND** tai Netscapen **EMBED** uudella **OBJECT**-elementillä. Objekti sisältää wav-formaatissa olevan biisin, jonka pitäisi alkaa soida heti kun sivu ladataan:

```

<OBJECT DATA="biisi.wav" TYPE="audio/wav">
<PARAM NAME="autostart" VALUE="true">
<PARAM NAME="hidden" VALUE="true">
</OBJECT>

```

Tämä oli vain esimerkki—useimpien ihmisten mielestä tällaiset häiriötekijät ovat äärimmäisen vastenmielisiä! En saanut esimerkkiä toimimaan omassa koneessani. Äänet kannattaa linkittää normaalilla tavalla (****), jolloin useimmat selaimet soittavat klikkaamisen jälkeen äänitiedoston automaattisesti käyttämällä pluginia tai jotain selaimen ulkopuolista ohjelmaa.

APPLET (vanhentunut)

- ◆ **code=tiedosto** (class-tiedosto)
- ◆ **codebase=URL** (class-tiedoston hakemisto)
- ◆ **name=merkkijono** (nimi sovelman sisäiseen käyttöön)
- ◆ **alt=merkkijono** (vaihtoehtoinen teksti)
- ◆ **width=pikselit** (sovelman leveys)
- ◆ **height=pikselit** (sovelman korkeus)
- ◆ **align=[top|middle|bottom|left|right]** (sovelman kohdistus sivulla)
- ◆ **hspace=pikselit** (tila sovelman vas. ja oik. puolella)
- ◆ **vspace=pikselit** (tila sovelman ylä- ja alapuolella)

APPLET on toistaiseksi varmempi tapa esittää Java-sovelmia; selainten tuki **OBJECT**-elementille on liian buginen. **APPLET** voi olla joko lohko- tai tekstielementtinä. **CODE** kertoo luokkatiedoston nimen ja **CODEBASE** luokkatiedoston hakemiston polun, jos tiedosto ei ole samassa hakemistossa kuin HTML-dokumentti. **CODE**-attribuutilla ei siis ilmoiteta hakemistoa, pelkkä tiedoston nimi. Attribuutit **WIDTH** ja **HEIGHT** ovat pakollisia. Jos selain ei tue Javaa tai Java-tuki on otettu pois päältä, selaimen pitäisi näyttää elementin sisältö:

```
<APPLET CODE="EuroCalc.class" CODEBASE="applets/"
WIDTH="220" HEIGHT="300" ALT="EuroCalc">
<P>Java-pluginia ei ole asennettu tai Java on kytketty pois
päältä selaimen asetuksista.</P>
</APPLET>
```

Huomaa että **CODEBASE**-attribuutilla ilmoitetun hakemiston pitää olla alihakemisto. Hakemistopuussa ylöspäin menevät linkit (**CODEBASE="../applets/"**) ovat turvallisuussyistä kiellettyjä.

FONT (vanhentunut)

- ◆ **color=väri** (fontin väri)
- ◆ **size=numero** (fontin koko)
- ◆ **face=fontti** (fontin nimi)

FONT-elementin käyttöä ei todellakaan voi suositella. CSS-tyylisäännöt ovat tehneet sen auttamattoman vanhanaikaiseksi. **FONT** aiheuttaa monia ongelmia, joista enemmän [Web-julkaiseminen](#)-sivulla. Väri ilmoitetaan joko RGB-arvona tai nimenä (ks. [Väreistä](#)-sivu). Koko ilmoitetaan asteikolla 1...7 tai suhteellisina arvoina, esim. **SIZE="+1"**. Peruskoko on 3. Voit ilmoittaa useampia fontteja. Jos ensimmäistä fonttia ei löydy koneesta, käytetään seuraavaa vaihtoehtoa ja niin edelleen. Esimerkki:

```
<FONT FACE="Arial,Helvetica,sans-serif" SIZE="4" COLOR="blue">
```

CSS tarjoaa enemmän mahdollisuuksia. Seuraavassa esimerkissä tekstikappaleiden fontti on tummanvihreä lihavoitu Arial, kappaleiden ensimmäisen rivin sisennys on 5% ja vasen marginaali on 4%. Toisen tekstikappaleityypin (**<P CLASS="alt">**) fonttikoko on 10 pistettä, kappaleen molemmat reunat tasataan, teksti ei ole lihavoitua mutta on kursivoitu ja alleviivattu. Vasen marginaali on suurempi kuin edellisessä tyyliässä.

```
P { font-family: Arial,Helvetica,sans-serif;
color: #006600;
font-weight: bold;
text-indent: 5%;
margin-left: 4% }
P.alt { font-size: 10pt;
text-align: justify;
font-weight: normal;
font-style: italic;
text-decoration: underline;
margin-left: 8% }
```

CSS-tyyliehdotuksissa säännöt usein *periytyvät* ylemmältä tasolta. Esimerkissämme `P.alt` perii ominaisuuksia normaalilta `P`-tekstikappaleelta, joten fontti on molemmissa kappaletyyleissä Arial, väri tummanvihreä ja ensimmäinen rivi sisennetään. `P`-kappaleen marginaali (4%) mitataan sivun vasemmasta reunasta eli `BODY`-elementin marginaalista, ja `P.alt`-kappaleen marginaali (8%) mitataan `P`:n marginaalista.

Fraasit

EM

`EM` tarkoittaa painotusta (emphasis). Selaimet esittävät sen yleensä kursivoituna. Koska `EM` liittyy dokumentin rakenteeseen, se on suositeltavampi vaihtoehto kuin kursivoiva `I`.

```
Herra Bushin <EM>Weltanschauung</EM> on muodostunut jossain syvällä.
```

Jos kirjoitan CSS-tyylisäännössä

```
p em { background-color: #FFFFCC;
      font-style: normal }
```

kaikki normaalien `P`-tekstikappaleiden sisällä olevat `EM`-elementit esitetään keltaisella taustalla normaalilla fonttityylillä (ei kursivoituna). Jos lukijan selaimessa ei ole CSS-tukea, `EM` näytetään normaalilla tavalla kursivoituna.

STRONG

`STRONG` tarkoittaa vahvaa painotusta. Selaimet esittävät sen yleensä lihavoituna. Koska `STRONG` liittyy dokumentin rakenteeseen, se on suositeltavampi vaihtoehto kuin lihavoiva `B`.

```
Ranskan ensimmäinen Bourbon-sukuinen hallitsija oli
<STRONG>Henrik IV.</STRONG>
```

CITE

`CITE` ei ole sitaatti (vrt. `Q`-elementti) vaan viittaus lähteeseen. Elementtiä käytetään teosten nimissä: kirjojen ja artikkelien nimet, elokuvien nimet, sävellysten nimet, maalausten nimet. Englanninkielisissä tekstissä myös sanoma- ja aikakauslehtien nimet ja laivojen nimet kuuluvat tähän kategoriaan. Selaimet esittävät `CITE`-elementin sisällä olevan tekstin yleensä kursivoituna, ja kuten arvaat, `CITE` on dokumentin rakenteeseen liittyvänä suositeltavampi kuin kursiivissa käytetty `I`-elementti. CSS:n avulla voi määritellä tarkemmin `CITE`-elementin ulkoasua. Esimerkkejä:

```
Picasson <CITE>Avignonin naiset</CITE> on kubismin merkkiteoksia.
James Joycen <CITE>Odysseus</CITE> ilmestyi vuonna 1922 Pariisissa.
Linux-koneita käytettiin <CITE>Titanic</CITE>-elokuvan tehosteiden
luomisessa.
```

CODE

Käytetään ohjelmakoodin esittämiseen. Selaimet esittävät **CODE**-tekstin tasavälisellä fontilla (esim. Courier). Kuten yleensä, selaimet päättävät itse rivinvaihtojen paikat. Jos haluat että rivinvaihdot, välilyönnit ja tabulaattorit säilytetään niin kuin ne ovat alkuperäisessä koodissa, laita **CODE**-tagit **PRE**-elementin sisälle.

```
<P>Näin olen viritellyt omassa koneessani Linuxin
komentokehoteen (ote on tiedostosta
<CODE>/etc/profile</CODE>):</P>

<PRE><CODE>
elif [ -n "$BASH_VERSION" ] ; then
    set -p
    if test "$UID" = 0 ; then
        PS1="\$(tty)-\w # "
    else
        PS1="\u@\$(tty)-\w > "
    fi
</CODE></PRE>
```

KBD

KBD-elementti sisältää käyttäjän tietokoneelle syöttämää tekstiä. Myös ohjelmien nimet (ls, pine, fetchmail jne.) kuuluvat joidenkin mielestä tähän kategoriaan silloin, kun kontekstiin liittyy ohjelman nimen kirjoittaminen komentokehoteessa. Selaimet esittävät **KBD**-tekstin yleensä tasavälisellä fontilla. Esimerkki:

```
Kun lopetat Linux-session, kirjoita komentokehoteessa
<KBD>exit</KBD>.
```

Ja tältä se näyttää selaimessasi:

Kun lopetat Linux-session, kirjoita komentokehoteessa `exit`.

SAMP

Esimerkkituloste **SAMP** (sample output) tarkoittaa tietokoneen sinulle heittämiä herjoja ja muita vastaavia ”tulosteita”. Selaimet esittävät sen yleensä tasavälisellä fontilla.

```
Jos komennat <KBD>make love</KBD>, komentotulkki vastaa<BR>
<SAMP>make: *** No rule to make target 'love'. Stop.</SAMP>
```

Selaimesi renderoi esimerkin näin:

```

Jos komennat make love, komentotulkki vastaa
make: *** No rule to make target 'love'. Stop.

```

Vastaus kiteyttää osuvasti tyypillisen nörtin elämäntilanteen.

VAR

VAR (variable) tarkoittaa mitä tahansa muuttujaa, esimerkiksi ohjelmakoodin argumenttia. Fysiikan ja matematiikan muuttujille sopii ehkä paremmin **i**-elementti. Selaimet renderoivat **VAR**-elementin sisällä olevan tekstin yleensä kursivoituna.

```

Linuxin kernelit ilmestyvät versionumerolla
<VAR>x</VAR>.<VAR>y</VAR>.<VAR>z</VAR>, jossa parillinen
<VAR>y</VAR> tarkoittaa &#8221;julkaisuversiota&#8221; ja
pariton <VAR>y</VAR> &#8221;kehitysversiota&#8221;.

```

Selaimesi näyttää esimerkin luultavasti kursiivilla:

Linuxin kernelit ilmestyvät versionumerolla *x.y.z*, jossa parillinen *y* tarkoittaa ”julkaisuversiota” ja pariton *y* ”kehitysversiota”.

ABBR ja ACRONYM (HTML 4.0)

ABBR (abbreviation) on lyhenne, sanan tai fraasin lyhennetty muoto. Inc., Ltd., Mr., et al., etc., jne., mm., esim. ja ts. ovat lyhenteitä. **ACRONYM** on muodostettu useamman sanan ensimmäisistä tai muutamasta ensimmäisestä kirjaimesta: WWW, ADSL, BBC, EU, YK, UNICEF, NATO, EMU. HTML 4 -suositus [ei selitä kovin tarkasti](#) näiden kahden eroa. Suositus näyttää noudattavan yllä olevaa tulkintaa yhdessä tekstikappaleessa, mutta esimerkeissä on käytetty toista tulkintaa.

Toisen tulkinnan mukaan **ACRONYM** on muuten sama asia kuin **ABBR** paitsi että se voidaan lausua yhtenä sanana. Toisin kuin normaali lyhenne joka lausutaan kirjain kirjaimelta (WWW, ADSL, BBC, EU, YK), akronyymi eli sanalyhenne voidaan lausua kokonaisena sanana (UNICEF, NATO, EMU). [Web Design Groupin](#) HTML-opas on tällä kannalla.

Selainten tuki näille elementeille on puutteellinen. IE 6 tunnistaa ainoastaan **ACRONYM**-elementin. Tämä voi olla riittävä syy unohtaa **ABBR**.

```

Lucasfilm <ABBR TITLE="Limited">Ltd.</ABBR> on tuottanut
paljon massaviihdettä.
<ACRONYM TITLE="Asymmetric Digital Subscriber Line">ADSL</ACRONYM>
kilpailee kaapelimodeemiyhteyksien kanssa.
<ACRONYM TITLE="Wireless Application Protocol">
WAP</ACRONYM>-kännykät tulivat markkinoille vuonna 1999.

```

TITLE-attribuutti voidaan lisätä melkein mihin tahansa elementtiin, jolloin se toimii samalla tavalla kuin kuvien **ALT**-tekstit joissain selaimissa: kun hiiren osoitin osuu kohdalle, näytetään ”tooltip”. Voit testata selaintasi viemällä kursorin alla olevan esimerkin lyhenteiden päälle:

Lucasfilm Ltd. on tuottanut paljon massaviihdettä.
ADSL kilpailee kaapelimodeemiyhteyksien kanssa.
WAP-kännykät tulivat markkinoille vuonna 1999.

Jos selaimessasi on kunnollinen CSS-tuki, lyhenteiden alla pitäisi näkyä yhden pikselin pisteviivalla tehty alleviivaus ja kursorin vieressä pitäisi näkyä kysymysmerkki. **ACRONYM**-elementissä fonttityylin pitäisi olla kapiteeli. Kapiteelin nimi on CSS:ssä **small-caps**. CSS-tyylisääntö on tässä:

```
abbr, acronym { border-bottom: 1px dotted black;
                  cursor: help }
acronym { font-variant: small-caps }
```

Fonttityylit

TT

TT (teletype) tarkoittaa tasavälistä fonttia (esim. Courier New tai Courier). Suositeltavampi tapa on käyttää dokumentin rakenteeseen liittyviä elementtejä: **CODE**, **SAMP** ja **KBD** viittaavat tiettyyn merkitykseen, kun taas **TT** ei liity mihinkään muuhun kuin fonttityyliin.

```
Kun lopetat Linux-session, kirjoita komentokehotteessa
<TT>exit</TT>.
```

B

Lihavoi tekstin. **STRONG** on suositeltavampi vaihtoehto voimakkaaseen korostamiseen. **B**-elementtiä voi kuitenkin käyttää esim. fysiikassa vektorien merkitsemiseen.

```
Ranskan ensimmäinen Bourbon-sukuinen hallitsija oli <B>Henrik IV.</B>
```

I

Kursivoi tekstin. Elementtiä voi käyttää tieteellisissä nimissä, sitaattilainoissa ja muissa ilmaisuissa, jotka vakiintuneen tavan mukaan esitetään kursiivilla (esim. suureiden symbolit kuten *T* fysiikassa). **EM**, **CITE** ja **VAR** esitetään yleensä myös kursiivilla. Käytä niitä normaaliin korostamiseen (**EM**), lähdeviittauksiin ja teosten nimiin (**CITE**) sekä muuttujiin (**VAR**).

```
Suden tieteellinen nimi on <I>Canis lupus</I>.
```

BIG ja SMALL

BIG- ja **SMALL**-elementtejä voidaan käyttää silloin, kun teksti halutaan esittää vähän isommalla tai pienemmällä fontilla kuin muu teksti. **SMALL** sopii hyvin esimerkiksi alaviitteisiin, osoitetietoihin ja tekijänoikeuksien ilmoittamiseen.

```
Normaalia tekstiä. <SMALL>Pienempää tekstiä.</SMALL>
Normaalia tekstiä. <BIG>Suurempaa tekstiä.</BIG>
```

Selaimesi näyttää esimerkin näin:

Normaalia tekstiä. Pienempää tekstiä. Normaalia tekstiä. Suurempaa tekstiä.

CSS-tyylisäännöissä vastaavien rakenteiden nimet ovat **larger** ja **smaller**:

```
p.copyright { font-size: smaller }
```

SUB

Tason pudotus, jolla voidaan esittää esimerkiksi alaindeksit. Selain renderoi **SUB**-elementin sisällä olevan tekstin pienikokoisella fontilla, joka sijoittuu vähän rivin alapuolelle. Esimerkki:

```
Metadonia (C<SUB>21</SUB>H<SUB>27</SUB>ON)
käytetään heroinistien vieroitushoidossa.
```

Selaimesi näyttää sen näin:

Metadonia (C₂₁H₂₇ON) käytetään heroinistien vieroitushoidossa.

SUP

Tason korotus, jolla voidaan esittää esimerkiksi yläindeksit. Selain renderoi **SUP**-elementin sisällä olevan tekstin pienikokoisella fontilla, joka sijoittuu vähän rivin yläpuolelle. Esimerkki:

```
M<SUP>lle</SUP> Marceau käveli 5<SUP>th</SUP> Avenuen
vasenta laitaa ajatellen kaavaa E = mc<SUP>2</SUP>.
```

Selaimesi näyttää sen näin:

M^{lle} Marceau käveli 5th Avenuen vasenta laitaa ajatellen kaavaa $E = mc^2$.

Kuten huomaat, typografinen laatu ei ole paras mahdollinen. ISO Latin-1 -merkistö sisältää merkit ¹, ² ja ³. Ne voidaan HTML-koodissa merkitä [symbolisilla notaatioilla](#) `¹`, `²` ja `³`. Tuloksena on siistimpi ulkoasu: $E = mc^2$. Eksponentit voidaan esittää myös korvaavilla tavoilla: $3^4 = 81$ tai $3^{*4} = 81$.

U (vanhentunut)

Alleviivatun tekstin esittämiseen. Käytä mieluummin CSS-tyylisääntöä vaikka tällä tavalla:

```
h3 { text-decoration: underline }
```

joka on parempi vaihtoehto kuin

```
<H3><U>Alleviivattu otsikko</U></H3>
```

Linkit ovat yleensä alleviivattuja. Jos käytät paljon alleviivausta normaalissa tekstissä, lukija voi sekoittaa alleviivatut sanat linkkeihin.

S ja STRIKE (vanhentuneet)

S ja **STRIKE** tarkoittavat tekstiä, jonka yli on vedetty vaakaviiva. **STRIKE** on selaimissa jonkin verran paremmin tuettu. **S** otettiin jostain ihmeellisestä syystä mukaan HTML 4 -suositukseen, vaikka identtinen **STRIKE** oli mukana jo HTML 3.2:ssa. Se siis tuotiin mukaan viralliseen HTML:ään ja samalla julistettiin ei-suositeltavaksi, vanhentuneeksi (*deprecated*). Esimerkki:

```
Tämä lause sisältää <STRIKE>yliviivatun</STRIKE> sanan.
```

CSS:n avulla sama efekti tehtäisiin näin:

```
.poistettu { text-decoration: line-through }
```

Tämän jälkeen yliviivatun sanan kohdalla voidaan käyttää **SPAN**-elementtiä:

```
Tämä lause sisältää <SPAN CLASS="poistettu">yliviivatun</SPAN> sanan.
```

Jos selaimessasi on CSS-tuki, yliviivauksen pitäisi toimia:

```
Tämä lause sisältää yliviivatun sanan.
```

Listat

UL

- ◆ `type=[disc|circle|square]` (luettelmapallon ulkoasu) (vanhentunut)

Tämän dokumentin alussa oleva sisällysluettelo on hyvä esimerkki **UL**-listasta (unordered list). **UL** pitää sisällään yhden tai useampia **LI**-elementtejä. Se ei voi olla **P**- tai **DT**-elementin sisällä, mutta voi olla minkä tahansa muun lohkoelementin sisällä. Sisäkkäiset listat ovat sallittuja: **UL**-listaan kuuluva **LI** voi pitää sisällään oman listansa. **TYPE**-attribuutin oletusarvona on yleensä **disc**, joka tarkoittaa pientä mustaa ympyrää; **TYPE="circle"** tekee ympyrän johon on piirretty pelkkä kehä; **TYPE="square"** tekee mustan pienen neliön. Esimerkki:

```
<UL>
  <LI>Klassinen
  <LI>Jazz
  <LI>Rock
</UL>
```

Selaimesi näyttää listan näin:

- Klassinen
- Jazz
- Rock

TYPE voidaan korvata CSS-säännöllä. Esimerkin toisella rivillä käytetään omaa kuvaa mustan ympyrän paikalla:

```
ul { list-style-type: disc }
ul.kuva { list-style-image: url(kuva.gif) }
```

OL

- ◆ `type=[1|a|A|i|I]` (järjestysluvun ulkoasu) (vanhentunut)
- ◆ `start=numero` (mistä numerosta aloitetaan) (vanhentunut)

OL on järjestysluvuilla esitetty lista. **OL** pitää sisällään yhden tai useampia **LI**-elementtejä. Se ei voi olla **P**- tai **DT**-elementin sisällä, mutta voi olla minkä tahansa muun lohkoelementin sisällä. Sisäkkäiset listat ovat sallittuja: **OL**-listaan kuuluva **LI** voi pitää sisällään oman listansa. **TYPE**-attribuutin oletuksena on **1** eli normaalit järjestysluvut (1., 2., 3. ja niin edelleen). Arvo **a** tekee luettelon pienillä kirjaimilla (a, b, c), **A** isoilla kirjaimilla (A, B, C), **i** pienillä roomalaisilla numeroilla (i, ii, iii) ja **I** isoilla roomalaisilla numeroilla (I, II, III).

```
<OL>
  <LI>Tilaa taksi
  <LI>Aja lentoasemalle
  <LI>Osta lippu
</OL>
```

Selaimesi näyttää listan näin:

1. Tilaa taksi
2. Aja lentoasemalle
3. Osta lippu

TYPE voidaan korvata CSS-säännöllä:

```
ol { list-style-type: decimal }
```

Mahdollisia arvoja CSS-standardissa ovat mm. **none** (ei mitään), **disc**, **circle** ja **square**, **decimal** (1., 2., 3.), **decimal-leading-zero** (01, 02, 03), **lower-roman** (i, ii, iii), **upper-roman** (I, II, III), **lower-latin** (a, b, c) ja **upper-latin** (A, B, C).

LI

- ◆ **type=[disc|circle|square]** (luettelmapallon ulkoasu) (vanhentunut)
- ◆ **value=numero** (järjestysluku OL-listan yhteydessä) (vanhentunut)

LI-elementti on listan alkio (list item). Lopputagi **** ei ole pakollinen. **LI** voi pitää sisällään sekä lohko- että tekstielementtejä. Voit siis laittaa **LI**-elementin sisälle tekstikappaleen, taulukon tai vaikka toisen listan:

```
<UL TYPE="disc">
  <LI>Windows
    <UL TYPE="circle">
      <LI>DOS
        <UL TYPE="square">
          <LI>Windows 95
          <LI>Windows 98
        </UL>
      <LI>NT
        <UL TYPE="square">
          <LI>Windows 2000
          <LI>Windows XP
        </UL>
    </UL>
  <LI>Mac OS
  <LI>Linux
</UL>
```

Selaimen pitäisi näyttää esimerkki näin:

- Windows
 - DOS
 - Windows 95
 - Windows 98
 - NT
 - Windows 2000
 - Windows XP
- Mac OS
- Linux

DL, DT ja DD

DL-elementti (definition list) on määrittelyjä sisältävä lista, joka pitää sisällään yhden tai useampia **DT**- ja **DD**-elementtejä. **DT** (definition term) sisältää määriteltävän termin ja **DD** (definition description) varsinaisen selostuksen. **DT** ei voi pitää sisällään lohkotason elementtejä (**P**, **TABLE**, **UL** jne.). **DD** voi sisältää lohkoelementtejä, myös sisäkkäisen **DL**-listan.

DD esitetään yleensä sisennettynä tekstinä. Lopputagit **DT**- ja **DD**-elementeille eivät ole pakollisia. Niitä kannattaa kuitenkin käyttää selainten bugien takia.

```
<DL>
  <DT><STRONG>Kontrapunkti</STRONG></DT>
    <DD>Kahden tai useamman itsenäisen melodian yhdistäminen
      musiikilliseksi kokonaisuudeksi. Polyfoninen sävellystapa,
      jossa pääteemaan liittyy itsenäisiä vastääniä.</DD>
  <DT><STRONG>Fuuga</STRONG></DT>
    <DD>Tärkein polyfoninen sävellysmuoto. Pääteema esiintyy yksin
      yhdessä äänessä, toinen teema jäljittelee sitä, edellinen
      ääni jatkuu säestävänä itsenäisenä vastateemana.</DD>
  <DT><STRONG>Sonaattimuoto</STRONG></DT>
    <DD>Esittely-, kehittely- ja kertausjaksoon jakautuva
      soitinmusiikin muototyyppi. Sarjallisten sävellysten
      yleinen ensimmäisen osan muotorakenne.</DD>
</DL>
```

Selaimesi näyttää esimerkin näin:

Kontrapunkti

Kahden tai useamman itsenäisen melodian yhdistäminen musiikilliseksi kokonaisuudeksi. Polyfoninen sävellystapa, jossa pääteemaan liittyy itsenäisiä vastääniä.

Fuuga

Tärkein polyfoninen sävellysmuoto. Pääteema esiintyy yksin yhdessä äänessä, toinen teema jäljittelee sitä, edellinen ääni jatkuu säestävänä itsenäisenä vastateemana.

Sonaattimuoto

Esittely-, kehittely- ja kertausjaksoon jakautuva soitinmusiikin muototyyppi. Sarjallisten sävellysten yleinen ensimmäisen osan muotorakenne.

DIR ja MENU (vanhentuneet)

DIR suunniteltiin alunperin hakemistolistauksiin, **MENU** menulistauksiin. Renderoinnin piti käsittääkseni olla erilainen kuin **UL**-listoissa, mutta koska selaimet esittävät näiden elementtien sisällön täsmälleen samalla tavalla kuin **UL**-listoissa, ne otettiin pois HTML 4 Strict -suosituksesta.

Taulukot

TABLE

- ◆ `width=[pikselit|prosentit]` (taulukon leveys)
- ◆ `border=pikselit` (solujen rajojen paksuus)
- ◆ `cellpadding=pikselit` (tila solujen sisällön ja rajan välissä)
- ◆ `cellspacing=pikselit` (tila solujen välissä)
- ◆ `frame=[void|above|below|hsides|lhs|rhs|vsides|box|border]` (taulukon ulkoraja)
- ◆ `rules=[none|groups|rows|cols|all]` (taulukon sisärajat)
- ◆ `summary=merkkijono` (yhteenvedo taulukosta)
- ◆ `align=[left|center|right]` (taulukon kohdistus sivulla) (vanhentunut)
- ◆ `bgcolor=väri` (taulukon taustaväri) (vanhentunut)

HTML-taulukko on **TABLE**-elementin sisällä. Se rakennetaan taulukkoriveistä (**TR**), joiden sisällä yksittäiset solut (**TD** ja **TH**) ovat. HTML 4 -suosituksessa taulukkorivit voidaan jakaa ryhmiin (**THEAD**, **TBODY** ja **TFOOT**) ja taulukon sarakkeet voidaan määrittellä **COL**- ja **COLGROUP**-elementeillä.

Taulukon ylemmässä rakenteessa **TABLE**-tagi on ensimmäisenä, sen jälkeen tulee **CAPTION** (taulukon ulkopuolelle sijoittuva otsikko), sitten **COL** ja/tai **COLGROUP**, niiden jälkeen **THEAD**, sitten **TBODY** ja viimeisenä **TFOOT**. Ainoastaan **TABLE** ja **TBODY** ovat pakollisia; viimeksi mainitun voi kuitenkin jättää kirjoittamatta, jos taulukossa ei ole **THEAD**- ja **TFOOT**-elementtejä. **TBODY** on automaattisesti mukana vaikka tageja ei koodissa näkyisikään. Alemmalla tasolla ovat varsinaiset taulukkorivit ja solut.

Taulukko ei voi olla **P**- tai **DT**-elementin sisällä, mutta voi olla minkä tahansa muun lohkoelementin sisällä. Sisäkkäiset taulukot ovat sallittuja, joten taulukko voi olla taulukon solun sisällä.

Yksinkertainen esimerkki:

```
<TABLE BORDER="1" CELLPADDING="3">
  <TR>
    <TD>Solu 1</TD>
    <TD>Solu 2</TD>
    <TD>Solu 3</TD>
  </TR>
  <TR>
    <TD>Solu 4</TD>
    <TD>Solu 5</TD>
    <TD>Solu 6</TD>
  </TR>
</TABLE>
```

Solu 1	Solu 2	Solu 3
Solu 4	Solu 5	Solu 6

WIDTH-attribuutti kertoo taulukon leveyden joko pikseleinä tai prosentteina. **BORDER** kertoo solujen reunusten paksuuden, **CELLPADDING** määrittelee solun sisällön ja solun rajan väliin jäävän tyhjän tilan leveyden ja **CELLSPACING** tarkoittaa solujen väliin jätettävää tyhjää tilaa. **SUMMARY** sisältää lyhyen yhteenvedon taulukosta; selainten tuki tälle attribuutille on aika huono, joten yhteenvedo kannattaa kirjoittaa normaaliin tekstikappaleeseen.

FRAME-attribuutilla voidaan määrätä mitkä taulukon ulkorajoista näytetään. Oletuksena on **FRAME="void"** jossa rajoja ei näytetä. Jos taulukon **BORDER**-attribuutissa on positiivinen arvo, oletuksena on **FRAME="border"** jossa näytetään kaikki rajat. Muut mahdolliset arvot ovat **box** (sama kuin **border**), **above** (vain taulukon yläreunus), **below** (alareunus), **vsides** (vasen ja oikea), **hsides** (ylä- ja alareunukset), **lhs** (vasen) ja **rhs** (oikea).

RULES määrittää mitkä taulukon sisärajoista näytetään. Jos taulukon attribuuttina on **BORDER="0"** tai **BORDER**-attribuuttia ei ole, oletuksena on **RULES="none"** (ei rajoja). Muut mahdolliset arvot ovat **groups** (vain sarakeryhmien ja riviryhmien rajat), **rows** (vain rivien rajat), **cols** (vain sarakkeiden rajat) ja **all** (kaikki rajat). Jos taulukon **BORDER**-attribuutissa on positiivinen arvo, oletuksena on **RULES="all"**.

CAPTION

- ◆ **align=[top|bottom|left|right]** (kohdistus) (vanhentunut)

CAPTION on varsinaisen taulukkorakenteen ulkopuolelle sijoittuva otsikko, joka voi pitää sisällään ainoastaan muita tekstielementtejä. Sijainti voidaan määrittellä vanhentuneella **ALIGN**-attribuutilla. Jos tätä elementtiä käytetään, sen pitää tulla heti **TABLE**-tagin jälkeen. Esimerkissä otsikko sijoitetaan taulukon yläreunan vasempaan osaan:

```
<TABLE BORDER="1" CELLPADDING="3">
  <CAPTION ALIGN="left"><STRONG>Otsikko</STRONG></CAPTION>
  <TR>
    <TD>Solu 1</TD>
    <TD>Solu 2</TD>
    <TD>Solu 3</TD>
  </TR>
  <TR>
    <TD>Solu 4</TD>
    <TD>Solu 5</TD>
    <TD>Solu 6</TD>
  </TR>
</TABLE>
```

Otsikko

Solu 1	Solu 2	Solu 3
Solu 4	Solu 5	Solu 6

TR

- ◆ `align=[left|center|right|justify]` (taulukkorivin solujen sisällön kohdistus)
- ◆ `valign=[top|middle|bottom]` (taulukkorivin solujen sisällön vertikaalinen kohdistus)
- ◆ `bgcolor=väri` (taulukkorivin taustaväri) (vanhentunut)

Taulukkorivi **TR** pitää sisällään varsinaiset taulukkosolut (**TD** ja **TH**). Rivi voi olla **THEAD**-, **TBODY**- tai **tfoot**-elementin sisällä. Yksinkertaisessa tapauksessa taulukkorivejä ei jaeta ryhmiin, jolloin **TR**-rivit ovat suoraan **table**-elementin sisällä.

TD

- ◆ `colspan=numero` (kuinka monta saraketta solu kattaa)
- ◆ `rowspan=numero` (kuinka monta riviä solu kattaa)
- ◆ `align=[left|center|right|justify]` (solun sisällön kohdistus)
- ◆ `valign=[top|middle|bottom]` (solun sisällön vertikaalinen kohdistus)
- ◆ `width=[pikselit|prosentit]` (solun leveys) (vanhentunut)
- ◆ `height=pikselit` (solun korkeus) (vanhentunut)
- ◆ `bgcolor=väri` (solun taustaväri) (vanhentunut)
- ◆ `nowrap` (älä tee rivinvaihtoja solun sisällä) (vanhentunut)

Taulukkosolu **TD** voi pitää sisällään sekä teksti- että lohkoelementtejä. Voit siis sijoittaa taulukon sisään tekstikappaleita, listoja, kuvia tai vaikka sisäkkäisiä taulukoita. **ALIGN** määrittelee solun sisällön kohdistuksen: **left** tasaa vasempaan reunaan, **right** oikeaan reunaan, **center** keskittää sisällön ja **justify** tasaa molemmat reunat. **VALIGN**-attribuutilla voidaan määrittellä vertikaalinen kohdistus. Oletuksena on **VALIGN="middle"**, joka voi vaikuttaa haitallisesti ulkoasuun silloin kun viereisessä solussa on enemmän tekstirivejä. **VALIGN="top"** korjaa tämän ongelman. Vanhentunut **nowrap** estää solun sisältämien tekstirivien jakamisen useammalle riville. Sen sijasta suositellaan käytettäväksi [symbolista notaatiota](#) ` `; (ns. kova välilyönti), joka laitetaan välilyöntien paikalle.

Esimerkissä on käytetty **rowspan**- ja **colspan**-attribuutteja, jotka laajentavat solun kattamaa aluetta rivien ja sarakkeiden suhteen:

```
<table border="1" cellpadding="3">
  <tr>
    <td rowspan="2" valign="top">Solu 1</td>
    <td colspan="2" align="center">Solu 2</td>
  </tr>
  <tr>
    <td>Solu 5</td>
    <td>Solu 6</td>
  </tr>
</table>
```

Selaimen pitäisi näyttää lopputulos näin:

Solu 1	Solu 2	
	Solu 5	Solu 6

TH

- ◆ `colspan=numero` (kuinka monta saraketta solu kattaa)
- ◆ `rowspan=numero` (kuinka monta riviä solu kattaa)
- ◆ `align=[left|center|right|justify]` (solun sisällön kohdistus; oletus "center")
- ◆ `valign=[top|middle|bottom]` (solun sisällön vertikaalinen kohdistus)
- ◆ `width=[pikselit|prosentit]` (solun leveys) (vanhentunut)
- ◆ `height=pikselit` (solun korkeus) (vanhentunut)
- ◆ `bgcolor=väri` (solun taustaväri) (vanhentunut)
- ◆ `nowrap` (älä tee rivinvaihtoja solun sisällä) (vanhentunut)

TH (table header cell) on taulukon otsikkosolu. Muilta osin se on täsmälleen samanlainen solu kuin **TD**. Selaimet näyttävät otsikkosolun tekstin yleensä lihavoidulla fontilla. Solun sisältö keskitetään eli oletuksena on `ALIGN="center"`. Normaalin taulukkosolun oletusarvona on `ALIGN="left"`. Esimerkki:

```
<TABLE BORDER="1" CELLPADDING="3">
  <TR>
    <TH>Otsikko 1</TH>
    <TH>Otsikko 2</TH>
    <TH>Otsikko 3</TH>
  </TR>
  <TR>
    <TD>Tekstiä solussa 4</TD>
    <TD>Tekstiä solussa 5</TD>
    <TD>Tekstiä solussa 6</TD>
  </TR>
</TABLE>
```

Otsikko 1	Otsikko 2	Otsikko 3
Tekstiä solussa 4	Tekstiä solussa 5	Tekstiä solussa 6

Jos selaimessasi on CSS-tuki, otsikkosoluja sisältävässä rivissä on vaaleansininen taustaväri, joka saadaan aikaan seuraavalla CSS-säännöllä:

```
<HEAD>
  <STYLE TYPE="text/css">
    th { background-color: #F9F9FF }
  </STYLE>
</HEAD>
```


COL (HTML 4.0)

- ◆ `span=numero` (sarakkeiden määrä)
- ◆ `align=[left|center|right|justify]` (solujen sisällön kohdistus)
- ◆ `valign=[top|middle|bottom]` (solujen sisällön vertikaalinen kohdistus)
- ◆ `width=[pikselit|prosentit]` (sarakkeen leveys)

COL-elementtejä voit käyttää taulukossa silloin, kun haluat määrittää jokaisen sarakkeen ominaisuudet erikseen CSS:n avulla. **COL**-elementit kirjoitetaan heti **TABLE**- ja **CAPTION**-tagien perään ennen taulukon varsinaista sisältöä. **COL** ja **COLGROUP** voivat esiintyä samassa taulukossa, jolloin **COL**-elementit ovat **COLGROUP**-elementtien sisällä. Lopputagi `</COL>` on kielletty. Seuraavassa CSS-esimerkissä taulukon kahden ensimmäisen sarakkeen taustaväri on vaaleanruskea, kolmas sarake on vaaleansininen ja viimeinen sarake on keltainen. **HEAD**-elementin sisällä on seuraavat CSS-tyylisäännöt:

```
col.brown { background-color: #FFF6ED }
col.blue  { background-color: #F9F9FF }
col.yellow { background-color: #FFF9DE }
```

COL-elementit kirjoitetaan heti taulukon alkuun:

```
<TABLE BORDER="1" CELLPADDING="3">
  <COL CLASS="brown" SPAN="2">
  <COL CLASS="blue">
  <COL CLASS="yellow">
  <TR>
    <TD>Solu 1</TD>
    <TD>Solu 2</TD>
    <TD>Solu 3</TD>
    <TD>Solu 4</TD>
  </TR>
  <TR>
    <TD>Solu 5</TD>
    <TD>Solu 6</TD>
    <TD>Solu 7</TD>
    <TD>Solu 8</TD>
  </TR>
</TABLE>
```

Selaimen pitäisi näyttää lopputulos näin:

Solu 1	Solu 2	Solu 3	Solu 4
Solu 5	Solu 6	Solu 7	Solu 8

COL- ja **COLGROUP**-elementtien yhteydessä ei voi käyttää kaikkia CSS-tyylisääntöjä. Säännöt periytyvät ylemmältä tasolta, joten taulukkosolu perii ominaisuuksia taulukkoriviltä ja taulukolta, ei sarakeelta. Sarakkeissa voi käyttää ainoastaan seuraavia CSS-ominaisuuksia: **background**, **width**, **visibility** ja **border**. Periytyvyysäännöstä johtuen monet selaimet (esim. Firefox) eivät huomioi **ALIGN**- ja **VALIGN**-attribuutteja **COL**- ja **COLGROUP**-elementtien yhteydessä, vaikka HTML-spesifikaatio mainitseekin kyseiset attribuutit.

COLGROUP (HTML 4.0)

- ◆ `span=numero` (sarakkeiden määrä; kielletty jos myös COL esiintyy)
- ◆ `align=[left|center|right|justify]` (solujen sisällön kohdistus)
- ◆ `valign=[top|middle|bottom]` (solujen sisällön vertikaalinen kohdistus)
- ◆ `width=[pikselit|prosentit]` (sarakeryhmän sarakkeiden leveys)

COLGROUP määrittelee *sarakeryhmiä*, **COL** määrittelee *yksittäisiä sarakkeita*. Jos esimerkiksi haluat kolmelle ensimmäiselle sarakkeelle yhteiset ominaisuudet tietyn asian suhteen, laitat sarakkeet **COLGROUP**-elementin sisään ja teet sarakeryhmälle CSS-tyylisäännöt. Ryhmän yksittäisiä sarakkeita voidaan edelleen muokata omilla **COL**-elementeillä, jotka laitetaan **<COLGROUP>**- ja **</COLGROUP>**-tagien väliin. Esimerkissä kolmen ensimmäisen sarakkeen taustaväri on vaaleansininen. Kahden seuraavan sarakkeen tausta on keltainen ja ryhmän ensimmäisen sarakkeen leveys on 30 prosenttia. **RULES**-attribuutin avulla selaimelle kerrotaan, että taulukon ulkorajojen lisäksi vain sarakeryhmien rajat näytetään. Ensin CSS-tyyli:

```
colgroup.blue { background-color: #F9F9FF }
colgroup.yellow { background-color: #FFF9DE }
```

COLGROUP- ja **COL**-elementit kirjoitetaan heti **TABLE**-tagin jälkeen:

```
<TABLE BORDER="1" CELLPADDING="4" CELLSPACING="0"
RULES="groups">
  <COLGROUP CLASS="blue">
    <COL>
    <COL>
    <COL>
  </COLGROUP>
  <COLGROUP CLASS="yellow">
    <COL WIDTH="30%">
    <COL>
  </COLGROUP>
  <TR>
    <TD>Solu 1</TD>
    <TD>Solu 2</TD>
    <TD>Solu 3</TD>
    <TD>Solu 4</TD>
    <TD>Solu 5</TD>
  </TR>
  <TR>
    <TD>Solu 6</TD>
    <TD>Solu 7</TD>
    <TD>Solu 8</TD>
    <TD>Solu 9</TD>
    <TD>Solu 10</TD>
  </TR>
</TABLE>
```

Selaimen pitäisi näyttää taulukko suunnilleen näin:

Solu 1	Solu 2	Solu 3	Solu 4	Solu 5
Solu 6	Solu 7	Solu 8	Solu 9	Solu 10

THEAD, TBODY ja TFOOT (HTML 4.0)

- ◆ `align=[left|center|right|justify]` (solujen sisällön kohdistus)
- ◆ `valign=[top|middle|bottom]` (solujen sisällön vertikaalinen kohdistus)

THEAD, **TBODY** ja **TFOOT** jakavat taulukon rivejä ryhmiin. Ne pitävät sisällään varsinaiset dataa sisältävät **TR**-taulukkorivit. Voit muokata CSS:n avulla jokaiselle riviryhmälle oman ulkoasun. Esimerkissä otsikkoriveille **THEAD** ja **TFOOT** määritellään kursivointi. Kahdelle ensimmäiselle taulukkoriville (`<TBODY CLASS="fin">`) annetaan vaaleanruskea taustaväri, seuraavat kolme riviä (`<TBODY CLASS="ger">`) ovat vaaleansinisiä ja kaksi viimeistä riviä (`<TBODY CLASS="latin">`) keltaisia. **RULES**-attribuutin avulla selaimelle kerrotaan, että taulukon ulkorajojen lisäksi vain sarakeryhmien ja riviryhmien rajat näytetään.

```
thead, tfoot { font-style: italic }
tbody.fin { background-color: #FFF6ED }
tbody.ger { background-color: #F9F9FF }
tbody.latin { background-color: #FFF9DE }
```

Jos samassa taulukossa on myös **COL**- tai **COLGROUP**-elementtejä, ne tulevat heti **TABLE**-tagin jälkeen. Niiden jälkeen on vuorossa taulukon varsinainen sisältö: **THEAD**, **TFOOT** ja **TBODY**, joista kukin pitää sisällään taulukkorivejä. Taulukossa voi olla useampia **TBODY**-elementtejä. Pienenä erikoisuutena on sääntö, jonka mukaan **TFOOT** pitää kirjoittaa HTML-koodissa ennen **TBODY**-elementtejä, vaikka sen sisältämät taulukkorivit esitetäänkin selaimessa viimeisenä **TBODY**-riviryhmien jälkeen:

```
<TABLE BORDER="1" CELLPADDING="4" CELLSPACING="0"
RULES="groups">
  <COLGROUP></COLGROUP>
  <COLGROUP SPAN="3"></COLGROUP>
  <THEAD>
    <TR>
      <TD>kieli</TD>
      <TD>1</TD>
      <TD>10</TD>
      <TD>100</TD>
    </TR>
  </THEAD>
  <TFOOT>
    ...rivi kuten yllä thead...
  </TFOOT>
  <TBODY CLASS="fin">
    <TR>
      <TD>suomi</TD>
      <TD>yksi</TD>
      <TD>kymmenen</TD>
      <TD>sata</TD>
    </TR>
    ...toinen taulukkorivi...
  </TBODY>
  <TBODY CLASS="ger">
    ...kolme taulukkoriviä...
  </TBODY>
  <TBODY CLASS="latin">
    ...kaksi taulukkoriviä...
  </TBODY>
</TABLE>
```

<i>kieli</i>	<i>1</i>	<i>10</i>	<i>100</i>
suomi	yksi	kymmenen	sata
viro	üks	kümme	sada
englanti	one	ten	hundred
saksa	eins	zehn	hundert
hollanti	eeenen	tiën	honderd
ranska	un	dix	cent
italia	uno	dieci	cento
<i>kieli</i>	<i>1</i>	<i>10</i>	<i>100</i>

Yhteiset attribuutit ja symboliset notaatiot

TITLE

TITLE-attribuutti voidaan lisätä melkein mihin tahansa elementtiin, jolloin selaimet voivat esittää lisäinformaatiota ”tooltip”-laatikossa. Tästä on hyötyä esimerkiksi **IMG**-, **OBJECT**-, **ACRONYM**- ja **ABBR**-elementtien yhteydessä. **TITLE** voidaan liittää myös normaaliin **A**-linkkiin, jolloin linkille saadaan oma ”tooltip”.

```
Vuonna 2006 keskusteltiin Suomen mahdollisesta
<ACRONYM TITLE="North Atlantic Treaty Organization">
NATO</ACRONYM>-jäsenyydestä.
```

Vie hiiren kursori ’NATO’-sanalle:

Vuonna 2006 keskusteltiin Suomen mahdollisesta NATO-jäsenyydestä.

LANG

LANG ilmoittaa käytetyn kielen:

```
Symbolisilla notaatioilla <SPAN LANG="en">(character entity,
entity reference)</SPAN> voidaan esittää ISO-8859-1-merkkejä.
```

LANG voidaan lisätä melkein mihin tahansa elementtiin. Kielikoodit ovat ISO 639 -standardin mukaisia: suomi **fi**, ruotsi **sv**, englanti **en**, ranska **fr**, saksa **de** ja niin edelleen. **LANG**-attribuuttia voidaan käyttää esimerkiksi **BLOCKQUOTE**-elementin yhteydessä, jos lohkositaatin teksti on eri kieltä kuin ympäröivän tekstin kieli. **Q**-elementin yhteydessä selain voi päätellä **LANG**-attribuutin arvon perusteella käytetäänkö kokolainausmerkkejä vai puolilainausmerkkejä ja miten toimitaan sisäkkäisissä lainauksissa. CSS-tyylisäännöillä voit esimerkiksi ehdottaa, että kaksikielisen dokumentin toinen kieli näytetään kursiivilla:

```
p[lang="en"] { font-style: italic }
span[lang="en"] { font-style: italic }
```

Huomaa että **a**-elementissä linkin kohdesivun kieli ilmoitetaan **HREFLANG**-attribuutilla. **LANG**-attribuuttia voidaan käyttää linkkitekstin kielen ilmoittamiseen.

```
<A HREF="http://www.paris.org/" HREFLANG="en" LANG="fr">
Les Pages de Paris</A> sisältää tietoja Pariisista.
```

Koko dokumentin kieli voidaan ilmoittaa dokumentin alussa olevassa **HTML**-tagissa:

```
<HTML LANG="fi">
```

tai **HEAD**-osuuden sisällä:

```
<HEAD>
<META HTTP-EQUIV="Content-Language" CONTENT="fi">
</HEAD>
```

CLASS, ID ja STYLE

Nämä attribuutit liittyvät CSS-tyylisääntöihin. HTML-elementille voidaan luoda erilaisia tyyli-versioita **CLASS**-attribuutin avulla. Voit esimerkiksi käyttää sivulla normaalin **p**-tekstikappaleen lisäksi kursivoitua kappaletyyliä, jolloin kirjoitat seuraavan tyyliehdotuksen:

```
<HEAD>
  <STYLE TYPE="text/css">
    p.muntyyli { font-style: italic }
  </STYLE>
</HEAD>
```

Tämän jälkeen kaikki **p**-elementit jotka alkavat tagilla

```
<P CLASS="muntyyli">
```

näytetään kursivoituna. **ID** eroaa **CLASS**-attribuutista siten, että se viittaa *vain yhteen* dokumentin kohtaan, kun taas **CLASS** määrittelee yhteisen tyylin *kaikille* samaa **CLASS**-attribuutin arvoa käyttäville elementeille. Kahdella elementillä ei siis voi olla samaa **ID**-arvoa. Seuraavassa esimerkissä kuva merkataan omalla **ID**-nimellä. Ensin luodaan kuvalle oma tyyli:

```
img#garfield { border: 3px solid red }
```

Kuva saa kolmen pikselin punaisen reunuksen. Kuva esitetään dokumentin **BODY**-osuudessa näin:

```
<IMG ID="garfield" SRC="kissa.jpg"
WIDTH="320" HEIGHT="457" ALT="[Garfield]">
```

ID-attribuuttia voidaan käyttää myös sivun sisäisissä linkeissä. Esimerkin kuvaan voitaisiin tehdä linkki sivun toisesta osasta:

```
<A HREF="#garfield">
```

Kaikki selaimet eivät tue **ID**-linkkejä, joten toistaiseksi kannattaa käyttää perinteistä ``-identifiointia linkin kohteen yhteydessä.

Kaikkia tyyliehdotuksia ei tarvitse kirjoittaa **HEAD**-osuuden sisällä olevaan **STYLE**-elementtiin. Sivun **BODY**-osuuden sisällä oleviin elementteihin voidaan liittää ”inline-tyylisääntö”:

```
<IMG STYLE="border: 3px solid red" SRC="kissa.jpg"
WIDTH="320" HEIGHT="457" ALT="[Garfield]">
```

STYLE on tässä siis attribuuttina. En suosittele inline-tyyliä, dokumentin ylläpito voi muuttua turhan hankalaksi. Jos inline-tyylejä käytetään dokumentissa, **HEAD**-elementin sisällä pitää ilmoittaa käytetty tyylikieli:

```
<HEAD>
<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">
</HEAD>
```

DIR

DIR-attribuutti kertoo tekstin suunnan. Oletus on vasemmalta oikealle (**DIR="ltr"**). Eräitä kirjoitusjärjestelmiä kirjoitetaan oikealta vasemmalle (**DIR="rtl"**). Jos dokumentissa on heprean-, arabian- tai persiankielistä tekstiä, tekstin suunta voidaan vaihtaa **DIR**-attribuutilla, joka kirjoitetaan esimerkiksi **P**-, **DIV**- tai **BLOCKQUOTE**-elementin yhteyteen.

```
<P>...länsimaista tekstiä...</P>
<P DIR="rtl">...hepreankielistä tekstiä...</P>
```

Tämän merkkauksen jälkeen esimerkin jälkimmäinen tekstikappale virtaa oikealta vasemmalle ja kappale on tasattu sivun oikeaan reunaan:

Seuraavan kappaleen suunta on toivottavasti oikealta vasemmalle.

אהר צהריים טובים

[Unicode](#) käyttää kaksisuuntaista algoritmia, jota käytetään aina kun dokumentissa on oikealta vasemmalle kirjoitettavia merkkejä. Yleensä lopputulos on OK, mutta joissain tapauksissa merkkien suunta jää neutraaliksi, jolloin tarvitaan **DIR**-attribuuttia. Esimerkiksi suomenkielinen lause, joka sisältää hepreankielisen lauseen, joka sisältää suomenkielisen lainauksen, vaatii **DIR**-attribuutin kertomaan hepreankielisen lauseen suunnan. Hepreankielinen lause (mukaan luettuna suomalainen lainaus) pitäisi kirjoittaa ``- ja ``-tagien väliin.

Symboliset notaatiot

Symbolisilla notaatioilla eli merkkientiteeteillä (*character entity, entity reference*) voidaan esittää kaikki ISO Latin-1 -merkkistön koodipaikoille 160–255 sijoittuvat merkit. Jos et muista merkin desimaalikoodia, käytä symbolista notaatiota. Muutamia esimerkkejä: Ä on `Ä`, è on `è`, © on `©`. (Samat merkit kirjoitetaan numeerisilla viittauksilla `Ä`, `è` ja `©`.) Jos dokumentin enkoodauksena on ISO-8859-1, voit syöttää koodipaikkojen 160–255 merkit myös suoraan—jos selain tunnistaa `Ä`-notaation isoksi Ä:ksi, se ymmärtää myös pelkän Ä-merkin HTML-koodissa.

[Merkistöistä](#)-sivulta löytyy kaikki ISO Latin-1 -merkkistön desimaalikoodit ja symboliset notaatiot. Mukana on myös tietoa Windows-merkkistön ”kiellettyistä merkeistä” ja Unicode-merkkistöstä. HTML 4 -suosituksessa dokumenttien merkkivalikoimaa on laajennettu käsittämään kaikki Unicode-merkit, ja symbolisten notaatioiden valikoima on vastaavasti kasvanut. Esimerkiksi neliöjuuren symboli voidaan kirjoittaa HTML-koodiin desimaalikoodilla `√` tai helpommin muistettavalla entiteetillä `√`.

ASCII-merkeillä `&`, `<` ja `>` on HTML-koodissa oma erikoismerkityksensä. Ongelmien välttämiseksi ne pitää aina kirjoittaa symbolisilla notaatioilla `&`, `<` ja `>`. Toinen vaihtoehto on kirjoittaa ASCII-koodi suoraan: `&`, `<` ja `>`.

ASCII	merkki	HTML
38	&	<code>&amp;</code>
60	<	<code>&lt;</code>
62	>	<code>&gt;</code>

Graafiset HTML-editorit saattavat korvata lainausmerkit `"`-notaatiolla. Tämä ei ole välttämätöntä. Poikkeuksena on kuitenkin tilanne, jossa attribuutin arvossa (esimerkiksi kuvan `ALT`-tekstissä) esiintyy kaksinkertaisia lainausmerkkejä.

Symbolinen notaatio ` ` tarkoittaa ns. sitovaa välilyöntiä (*no-break space, non-breaking space*). Rivinvaihto voidaan estää erottamalla kaksi sanaa välilyönnin sijasta ` `-notaatiolla. Tyypillisiä esimerkkejä:

<code>12&nbsp;m</code>	<code>12 m</code>
<code>HTML&nbsp;4.0</code>	<code>HTML 4.0</code>
<code>J.R.R.&nbsp;&nbsp;Tolkien</code>	<code>J.R.R. Tolkien</code>
<code>5&nbsp;000&nbsp;000</code>	<code>5 000 000</code>

Sitovaa välilyöntiä voidaan käyttää myös ylimääräisen tyhjän tilan luomiseen. CSS on kuitenkin useimmissa tapauksissa parempi vaihtoehto. Kappaleen ensimmäinen rivi voidaan sientää käyttämällä neljää tai useampaa ` `-merkkiä, mutta parempi tapa on CSS:n `text-indent`. Rivinvaihdot estetään taulukon solun sisällä käyttämällä ` `-merkkiä sanojen välissä, mutta suositeltavampi tapa on CSS:n `white-space: nowrap`.

Huomautuksia

Lohkoelementit ja tekstielementit

Lohkotason elementit (*block-level elements*) voivat pitää sisällään tekstielementtejä (*inline elements*) ja muita lohkoelementtejä. Ainoat poikkeukset ovat **P** ja **DT**, jotka eivät voi pitää sisällään muita lohkoelementtejä. Lohkoelementit aiheuttavat yleensä rivinvaihdon.

Seuraavat elementit ovat lohkoelementtejä:

```

    otsikot:      H1 H2 H3 H4 H5 H6
tekstikappaleet: P* PRE ADDRESS BLOCKQUOTE NOSCRIPT
    osiin jakavat: HR DIV CENTER
    listat:      OL UL DL DT* DIR MENU
    taulukko:    TABLE
  
```

Myös seuraavia elementtejä voidaan pitää lohkoelementteinä, koska ne voivat pitää sisällään muita lohkoelementtejä:

```

    taulukoissa: TR TD TH THEAD TBODY TFOOT
    listoissa:  LI DD
  
```

Seuraavat elementit voivat olla joko lohkoelementtejä tai tekstitason elementtejä:

```

    OBJECT MAP APPLETT SCRIPT
  
```

Jos niitä käytetään tekstielementteinä, ne eivät voi pitää sisällään lohkotason elementtejä. Tekstielementit pitävät yleensä sisällään tekstiä tai muita tekstielementtejä. Viimeiseksi lista tekstielementeistä:

```

    geneeriset:  A BR SPAN FONT Q
                SCRIPT* OBJECT* APPLETT*
    kuvat:      IMG MAP*
    taulukoissa: CAPTION
    fraasit:    EM STRONG ABBR ACRONYM
                CITE VAR CODE KBD SAMP
    fonttityylit: B I TT BIG SMALL SUB SUP
                 U S STRIKE
  
```

Alku- ja lopputagit

Sulje HTML-elementti aina lopputagilla (esim. `<h1>...</h1>`). Lopputagi on kuitenkin *kielletty* seuraavissa elementeissä:

```

    IMG HR BR AREA PARAM COL META
    INPUT FRAME ISINDEX LINK BASE BASEFONT
  
```

Seuraavista elementeistä lopputagin voi jättää pois:

```

    LI DT DD OPTION P TR TD TH COLGROUP THEAD TFOOT
  
```


Sekä alku- että lopputagin *voi* jättää pois seuraavista elementeistä:

HTML HEAD BODY TBODY

Olen kirjoittanut varmuuden vuoksi aina kaikki lopputagit. Jossain vaiheessa voit kiinnostua [XHTML](#)-kielestä. XHTML:ssä lopputagit ovat pakollisia, elementit ja attribuutit kirjoitetaan aina pienillä kirjaimilla, ja attribuuttien arvot pitää kirjoittaa poikkeuksetta lainausmerkkien sisään. Jos sivusi ovat näiltä osin jo nyt kunnossa, välttyt massiiviselta uudelleenkirjoittamiselta siinä vaiheessa kun muunnat sivujasi XHTML-muotoon.

Vanhentuneet ja uudet elementit

Nämä elementit ovat vanhentuneita (*deprecated*) eikä niitä suositella:

**CENTER APPLET ISINDEX FONT BASEFONT
U S STRIKE DIR MENU**

Nämä elementit eivät olleet mukana vanhemmassa [HTML 3.2](#) -suosituksessa:

**NOSCRIPT SPAN Q OBJECT PARAM ABBR ACRONYM S
COL COLGROUP THEAD TBODY TFOOT
BDO BUTTON DEL FIELDSET FRAME FRAMESET IFRAME
INS LABEL LEGEND NOFRAMES OPTGROUP**

Isot ja pienet kirjaimet

HTML-elementit ja niiden attribuutit ovat ”case-insensitive”: on aivan sama käytätkö isoja vai pieniä kirjaimia. Suosittelen kuitenkin pienten kirjainten käyttöä. Tässä dokumentissa on käytetty isoja kirjaimia ainoastaan luettavuuden takia.

Isot ja pienet kirjaimet attribuuttien arvoissa

Attribuuttien arvot kannattaa aina kirjoittaa pienillä kirjaimilla. Tässäkin säännössä on poikkeuksia, esim. arvot **left**, **right**, **center** ja muut samantyyppiset ennalta määrättyihin vaihtoehtoihin kuuluvat sanat, mutta varminta on käyttää aina pieniä kirjaimia.

Lainausmerkit attribuuttien arvoissa

Attribuutin arvon ei tarvitse olla lainausmerkkien sisällä, jos arvo ei sisällä muita merkkejä kuin kirjaimet a–z ja A–Z, numerot 0–9, piste, väliviiva, alaviiva () ja kaksoispiste. Tästä huolimatta käytä aina lainausmerkkejä—kannattaa muodostaa itselleen yhtenäinen käytäntö, jolloin ei tule ongelmia eikä tarvitse muistaa poikkeuksia.

Jos lainausmerkkejä käytetään lainausmerkkejä sisältävän arvon ympärillä, käytä kaksinkertaisia ASCII-lainausmerkkejä (") arvon ympärillä ja yksinkertaisia lainausmerkkejä (')

arvon sisällä (tai toisinpäin). Arvon sisällä saa käyttää myös symbolista notaatiota `"`; tai desimaalikoodia `"`; . Valitse mieltyttävien vaihtoehto:

```
<IMG SRC="bobby.jpg" ALT="De Niro elokuvassa 'Taksikuski'">
<IMG SRC="bobby.jpg" ALT='De Niro elokuvassa "Taksikuski"'>
<IMG SRC="bobby.jpg" ALT="De Niro elokuvassa &quot;Taksikuski&quot;">
<IMG SRC="bobby.jpg" ALT="De Niro elokuvassa &#34;Taksikuski&#34;">
```

Attribuutin arvon merkkijonoa ei saa jakaa useammalle riville.

Ongelmalliset `<` ja `>` ja `&`

HTML-kielen syntaksista johtuen merkit `<`, `>` ja `&` pitää aina esittää symbolisilla notaatioilla `<`; , `>`; ja `&`; (myös `PRE`-elementin sisällä).

URL-osoitteet

URL-osoitteiden pitää olla lainausmerkkien sisällä (esimerkiksi `HREF="http://www.isp.fi/munsivu.html"`). Poikkeuksena ovat suhteelliset osoitteet (`HREF=munsivu.html`), mutta suosittelen niissäkin lainausmerkkien käyttöä. URL-osoitteiden sallitut merkit ovat kirjaimet a–z ja A–Z, numerot 0–9 ja merkit `-_~.!*'()`. Lisäksi voidaan tietyissä tapauksissa käyttää näitä URL-osoitteiden rakenteeseen liittyviä merkkejä: `;/?:@=&#+$`

Jos haluat että sivu menee validaattoreista läpi, kirjoita `&`-merkki URL-osoitteessa aina `&`-notaatiolla. Selaimet voivat luulla `&`-merkkiä merkkientiteetin aloitukseksi.

Välilyönti ja tilde URL-osoitteessa

Välilyönti pitää URL-osoitteessa ilmoittaa merkkiyhdistelmällä `%20`. (Välilyönnin ASCII-desimaalikoodi on 32, heksadesimaalikoodi on 20.) Tilde `~` pitäisi ilmoittaa merkkiyhdistelmällä `%7E`, joskin uudemmat standardit joustavat tilden kohdalla. Välilyönti ei ole sallittu myöskään seuraavassa tapauksessa, jossa sivulla on lukijoiden antamaan palautteeseen tarkoitettu sähköpostilinkki:

```
<A HREF="mailto:tunnus@isp.fi?subject=Palautetta%20sivustasi">
```

Sähköpostilinkissä on valmis aihe ”Palautetta sivustasi”. Välilyönti korvataan `%20`-notaatiolla.

Kommentit

HTML-koodissa kommentti alkaa merkeillä `<!--` ja loppuu merkkeihin `-->`. Kommentin sisällä ei saa olla merkkijonoa `--` eli kahta tavuviivaa peräkkäin. CSS-kommentti on samannäköinen kuin C-ohjelmointikielessä: kommentti alkaa merkeillä `/*` ja loppuu merkkeihin `*/`.

ALT on pakollinen

Älä unohda kuvista **ALT**-tekstiä. Teksti on tarkoitettu vaihtoehtoiseksi sisällöksi lukijoille, jotka eivät jostain syystä näe kuvia tai ovat ottaneet selaimesta kuvien latauksen pois päältä. HTML 4 -suosituksessa **ALT** on pakollinen sekä normaaleissa **IMG**-kuvissa että kuvakarttojen (image map) **AREA**-elementeissä.

Selainvalmistajien omat lisäykset

tarkoittaa tässä Internet Explorerin ja Netscapen omia HTML-elementtejä ja -attribuutteja, jotka eivät kuulu W3C:n suositukseen. HTML-puristien mielestä niitä ei pitäisi käyttää. Olen samaa mieltä. Haluan että sivuni toimivat mahdollisimman samalla tavalla kaikille lukijoille. Alla oleva taulukko sisältää siis esimerkkejä siitä mitä ei pidä tehdä. Jos kaikesta huolimatta käytät taulukossa esitettyjä juttuja, varmista että sivu kokonaisuutena toimii myös muiden selainten käyttäjille.

Selain	Oma lisäys	HTML 4
IE	BODY -elementin attribuutit TOPMARGIN , LEFTMARGIN , RIGHTMARGIN ja BOTTOMMARGIN	[CSS]
IE	HR -elementin attribuutti COLOR	[CSS]
IE	TABLE -, TD - ja TH -elementtien attribuutit BACKGROUND , BORDERCOLOR , BORDERCOLORDARK ja BORDERCOLORLIGHT	[CSS]
IE	TR -elementin attribuutit BORDERCOLOR , BORDERCOLORDARK ja BORDERCOLORLIGHT	[CSS]
IE	THEAD -, TBODY - ja TFOOT -elementtien attribuutti BGCOLOR	[CSS]
IE	MARQUEE (vierivä teksti)	
IE	BGSOUND (taustaaäni)	OBJECT
Netscape	EMBED (esim. taustaaäni)	OBJECT
Netscape	BLINK (vilkkuva teksti)	[CSS]
Netscape	SPACER (tällä tehdään tyhjää tilaa)	[CSS]
Netscape	FONT -elementin attribuutit POINTSIZ e ja WEIGHT	[CSS]
Netscape	MULTICOL (sivun jako palstoihin)	TABLE tai CSS
Netscape	LAYER ja ILAYER (tasojen sijoittelu)	[CSS]
Netscape	NOBR (estää rivinvaihdon)	&nbsp; ;

Markku Immonen <wired.poet@sci.fi>

Viimeksi muokattu: 17.2.2006

URL: <http://www.saunalahti.fi/wpoet/fin/htmlref.htm>