



Drop tables til design, og få fuld udbytte af CSS!

Fremtidens hjemmeside-udvikling består af ganske lidt HTML, som er logisk opbygget, samt en masse CSS, som præsenterer siden på en pæn måde. Artiklen gennemgår fremgangsmetoden til opbygning af en side, ved brug af CSS.

Skrevet den **07. Feb 2009** af **hermandsen** | kategorien **(D)HTML / CSS** | ★★★★★

Historie

Da HTML blev udviklet for mange år siden, var det kun meningen at det skulle kunne fremvise tekst, så det stod overskueligt og struktureret. Den gang var der ikke noget der hed layout, men web-designerne skulle bruge noget så de også kunne lave en nogen lunde pæn hjemmeside, som ikke bare var en lang bane med tekst.

Tables var den gang svaret! Simpelthen fordi tables gjorde det muligt at opdele tingene i bokse, sætte en vis afstand mellem de forskellige objekter, gøre brug af rammer og så videre.

Det var tiden hvor man delte en hel side ind i et eller flere tables, og så lavede utallige row/colspan, indsatte tomme billeder for at få afstand, og du kender sikkert selv flere tricks!

Nu sidder du sikkert og siger: "Jamen alle de store sider er da lavet med tables. F.eks. www.dr.dk, www.computerworld.dk ja selv www.eksperten.dk!" og så svarer jeg: "Ja, men hvorfor ikke lære det rigtige før dem?" Når du er færdig med denne artikel vil du forstå! ;)

Standarder

Før vi fortsætter vil jeg gerne fortælle dig noget om åbne standarder! HTML og CSS er begge åbne standarder, hvilket betyder at alle kan:

- læse hvordan de bruges.
- gøre brug af dem.
- komme med forslag til forbedringer.

Men hvorfor er det så vigtigt med åbne standarder??

Prøv at forestille dig at internettet og HTML var blevet udviklet af to forskellige firmaer...

Et firma lavede deres browser, og det andet lavede deres. Begge fulgte de deres egen lukkede standard, og i sidste ende ville det resultere i at du skulle bruge en browser for at komme ind på nogle sider, og en anden til andre sider!! Forestil dig så at der var 3 forskellige firmaer, og forestil dig så igen at der var 7 forskellige firmaer!! I sidste ende skulle du sidde som bruger med 7 forskellige browsere og prøve dig frem for hver side du gik ind på, hvilken browser siden nu var lavet til!! Jeg tror næppe internettet var blevet så populært, hvis ikke der fra starten havde været en fælles åben standard som i dag!

Om 5-10 år sidder vi måske i vores små hjem og til den tid kan køleskabet, fjernsynet og komfuret gå på nettet, hvilket betyder at vi bliver nød til at have en standard som er tilgængelig for alle, og som alle kan gøre brug af! Det gør det også lettere for udviklerne at lave tingene ordentligt på den måde!

3,2,1 Start

Så går vi i gang. I denne artikel vil jeg udelukkende gøre brug af XHTML og CSS version 2. Grunden til det, er at de begge er ret godt fremtidssikrede, og det er hvad W3C (World Wide Web Consortium) anbefaler! Jeg antager desuden at du har lidt base-viden inden for HTML og CSS.

Så går vi ellers i gang:

Opret en mappe til dine filer, og derefter følgende:

- **index.html**
- **default.css**
- **/default** (endnu en mappe)

Åbn filerne i et rent tekst-program (gerne med syntax-highlighting, hvis du har). Notepad er fin, jeg synes selv godt om Programmeres Notepad 2 (<http://www.pnotepad.org/download.php>, husk at hente version 2).

Copy/paste dette ind i *index.html*, så gennemgår vi det lige hurtigt:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="da">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=ISO-8859-1" />
    <title>Drop tables til design!</title>
    <link rel="stylesheet" type="text/css" href="default.css"
media="screen,projection" title="Default" />
  </head>
  <body>
    <div id="banner">
      <h1><span>Se min flotte hjemmeside!</span></h1>
    </div>
    <div id="menu">
      <h2><span>Dette er hoved-menu'en</span></h2>
      <ul>
        <li><a href="index.html">Forside</a></li>
        <li><a href="index.html">Næste side</a></li>
        <li><a href="index.html">3. side</a></li>
        <li><a class="external"
href="http://www.eksperten.dk/">Eksperten</a></li>
        <li><a class="external" href="http://www.ebruger.dk/">eBruger</a></li>
      </ul>
    </div>
    <div id="content">
      Bla bla bla...
    </div>
  </body>
</html>
```

Fordi vi benytter XHTML skal vi starte med at fortælle browseren at dette er XML (som XHTML jo kommer af). Samtidig fortæller vi at det er XHTML version 1.1, dansk sprog, hvilke tegn der er bruges osv. Informationen er for så vidt ligegyldig for personen der skal se siden, men nødvendig for at browseren kan vise siden ordentligt!

Desuden fortæller vi at vi gerne vil bruge vores stylesheet, *default.css*, og at det kun kan bruges i forbindelse med fremvisning på en skærm, eller en projekter.

Resten af dokumentet er helt analogt. Vi har et banner, en menu, bestående af punkter og noget indhold (content). Alt samme inddelt i div-tags.

Hvis vi så siden i en browser nu, ville den se ualmindelig kedelig ud. Derfor skal vi nu igang med CSS'en...

Under *default*-mappen skal du oprette en fil med navnet *basic.css*. Åbn også den og skriv følgende:

```
body {
  font-family: Verdana, Tahoma, Arial, Helvetica, Sans-serif;
  font-size: 13px;
  background-color: white;
  color: black;
}
```

Hvis vi så resultatet i browseren nu, ville der ikke komme noget som helst, da vi jo refererer til *default.css* i samme mappe som *index.html* ligger i. Det gør vi noget ved.

Åbn *default.css* og skriv:

```
@charset "ISO-8859-1";
@import "default/basic.css";
```

Tadaaa... Resultatet ses med det samme, men hvad skete der lige??

Først fortalte vi hvilken type bogstaver der findes i dokumentet, **@charset "ISO-8859-1"**, hvilket ikke giver noget synligt resultat.

Men **@import "default/basic.css"** til gengæld:

En smart ting ved CSS, er at man kan importere andre CSS-dokumenter. På den måde kan du let holde tingene adskilt, eller f.eks. importere standard indstillinger, hvis du udvikler til et stort firma som har nogle standarder for hvordan deres overskrifter, links osv., skal se ud.

Lad os straks oprette endnu et CSS-dokument under *default*-mappen, *banner.css*. Vi kan ligeså godt smide den ind i *default.css* med det samme, så der nu står:

```
@charset "ISO-8859-1";
@import "default/basic.css";
@import "default/banner.css";
```

Åbn den, og lad os få placeret banneret. Vi starter med følgende:

```
#banner {
  position: absolute;
  left: 0px;
  top: 0px;
  height: 50px;
  width: 100%;
  margin: 0px;
  background-color: black;
  color: white;
}
```

Nu bliver det interessant!! Vi tager det en linie ad gangen:

1. #banner {

#-tegnet fortæller at der refereres til noget med en id-attribut i HTML'en. I dette tilfælde, objektet med id'et "banner" (det samme id må kun være at finde en gang på hver side, alternativt må man bruge class).

2. position: absolute;

Position fortæller hvordan elementet skal positioneres. **static** ignorerer positioneringen (standard).

relative sætter objektet i forhold til hvor det står. **absolute** sætter det præcist på pladsen, afhængig af afstanden fra kanterne. Og til sidst **fixed** som holder objektet fast på stedet, også når der scrolles (understøttes ikke af Internet Explorer).

3. left: 0px;

Sætter objektets venstre kant 0 pixels i forhold til venstre kant af indholdssiden.

4. top: 0px;

Sætter objektets top kant 0 pixels i forhold til toppen af indholdssiden.

Resten burde være til at gennemskue.

Se resultatet i din browser.

Hmmm... Det er næsten godt nok, men teksten kunne måske godt sættes lidt mere præcist.

Tilføj dette til *banner.css*:

```
#banner h1 {  
  padding: 6px 0px 6px 10px;  
  margin: 0px;  
}
```

Vi tager det endnu en gang, linie for linie:

1. #banner h1 {

Nu sætter vi egenskaber for alle de h1-tags der findes inde i #banner-id'et. Dette er intet mindre end genialt, da vi på den måde ikke behøver at give hver h1 en klasse eller et id, men blot kan referere til #banner-id'et, og finde alle h1-tags derfra!

2. padding: 6px 0px 6px 10px;

At opstille tingene på denne måde kaldes for "shorthand-properties". I stedet kunne vi have skrevet **padding-top: 6px; padding-right: 0px;** osv., men dette er hurtigere. Tallenes rækkefølge er: **top, right, bottom, left**. Altså start fra top, og så med uret rundt.

3. margin: 0px;

Igen en shorthand-property. I stedet for at fortælle hver eneste margin hvor lang den skal være, sætter vi dem alle sammen til 0px. Vi kunne også have skrevet **margin: 0px 0px 0px 0px;**, men dette er betydelig nemmere.

Næste trin er menu'en... Tilføj:

```
@import "default/menu.css";
```

til *default.css*, og opret filen *menu.css* i *default*-mappen.

Lad os starte med at placere den:

```
#menu {  
  position: absolute;  
  top: 50px;  
  left: 0px;  
  background-color: #88F;  
  color: black;  
  height: 350px;  
}
```

Her er tingene helt gennemskuelige! Hvis du undrer dig over måden **background-color** er skrevet på, så kan du læse mere på <http://www.eksperten.dk/artikler/80>.

Det næste vi skal, er at fjerne **margin** og **padding** fra ul-tag'et. En "Unordered List" (ul) indeholder som standard lidt margin, og lidt padding (desværre lidt afhængig af browseren).

I *menu.css* skriver vi derfor:

```
#menu ul {  
  margin: 0px;  
  padding: 0px;  
}
```

Hvis du har fulgt med, burde ovenstående ikke blive forklaret nærmere.

Hvad med vores h2-tag? Vi har efterladt noget tekst hvor i der står "Dette er hoved-menu'en", men egentlig har vi ikke brug for det!! Det er lavet så andre design-udviklere måske kunne gøre brug af det!! ;)

```
#menu h2 {  
  display: none;  
}
```

Så forsvandt det også!! Læg mærke til at der i HTML'en er skrevet **<h2>Dette er hoved-menu'en[/span]</h2>**. Det gør vi fordi hvis nu nogen skulle have lyst til at smide et billede ind over vores menu, så kunne han/hun bare sætte span-tag'et til **display: none;**, og så sætte h2's **background-image** egenskab til noget! Der gives et eksempel på det senere! ;)

Nu begynder vores side så småt at ligne noget... Dog skal den gøres meget pænere!! Vi smider også lige dette ind:

```
#menu a {  
  text-decoration: none;  
  color: black;  
  background-color: transparent;  
  width: 120px;
```

```
display: block;
}
```

Vi referer til alle a-tags under #menu-id'et, fjerner understregningen, sætter tekst-farven til sort, bruger en gennemsigtig baggrundsfarve, så farven fra menu-id'et træder i gennem, sætter en fast bredde og siger at tag'et skal vises som en blok. Sidst nævnte er vigtigt, da vi på den måde fortæller browseren at dette ikke bare er et element som skal fylde bredden/højden af sig selv, men at vi er med til at bestemme hvor stort elementet skal se ud.

Så skal vi ha' tingene til at se lidt smartere ud!

```
#menu a:hover {
  color: black;
  background-color: #66C;
}
```

Vi referer igen til alle a-elementer under menu-id'et, men denne gang sætter vi kun egenskaberne for det når det er **:hover**-aktivt. Mange tror godt nok at det kun er links der kan bruges hover på, men dette er ikke tilfældet! Måske skyldes det at det er endnu en funktion som Internet Explorer ikke understøtter, på andet end links! ;)

På links kan der kun bruges **:link** og **:visited**. Mens der på resten også kan bruges **:active**, **:hover** og **:focus**.

Vi går straks videre. Et par af de links vi har liggende er eksterne links, her angivet med klasse-navnet **external**. Dem ønsker vi at fremvise anderledes:

```
#menu a.external:hover {
  color: black;
  background-color: #44B;
}
```

Vi vælger menu-id'et, finder alle a-tags som en external-klasse og vælger kun hover-attributten. Av, vi er ved at være langt nede!! ;)

Nu ser det pludselig anderledes ud!

Det sidste vi skal have fat i er content-id'et.

Smid

```
@import "default/content.css";
```

Ind under *default.css*, og lad os så komme i gang:

```
#content {
  position: absolute;
```

```
top: 55px;
left: 125px;
right: 5px;
height: 337px;
overflow: auto;
border: 1px solid black;
padding: 3px;
}
```

Du burde kende det meste, `right`-egenskaben er dog ny. Denne sætter afstanden fra højre side af objektet til højre browser-kant (Internet Explorer er desværre lidt ringe til at finde ud af det. Det samme gælder **bottom**-egenskaben). En anden ny er `overflow`. `overflow` fortæller hvordan tingene skal gemmes af vejen. Denne kan være:

visible [standard, viser hele elementet]
hidden [gemmer ting som ikke kunne være inde i boksen]
scroll [laver scroll-bars lodret og vandret]
auto [laver scroll-bars, hvis der er brug for det]
inherit [arver egenskaben fra tidligere]

Den næste er `border`-egenskaben. Igen en shorthand-egenskab som er sat til **`border-width: 1px;`**, **`border-style: solid;`** og **`border-color: black;`**

Hvis vi havde en masse underelementer, kunne vi nemt formatere disse også, men der til burde du nu selv have viden nok til at fortsætte.

Som jeg lovede tidligere, skulle vi lave `<h2>Dette er hoved-menu'en[/span]</h2>`-tricket. Det kommer her:

Åbn `banner.css` og tilføj først:

```
#banner span {
  display: none;
}
```

Så forsvandt teksten først!

Det næste vi skal er at ændre:

```
#banner h1 {
  padding: 6px 0px 6px 10px;
  margin: 0px;
}
```

til

```
#banner h1 {
  padding: 6px 0px 6px 10px;
  margin: 0px;
  background-image:
url("http://hermandsen.homepage.dk/exp/droptables/default/img/banner.png");
  height: 38px;
  background-repeat: no-repeat;
}
```

Vi tager det lige lidt ad gangen:

4. background-image: url("http://hermandsen.homepage.dk/exp/droptables/default/img/banner.png");
Vi referer til et billede som skal ligge i under h1-tag'et. Man referer til billedet ligesom ved normal HTML som f.eks. **foo**.

5. height: 38px;

Hvorfor 38px? Fordi banner-id'et er 50px, og vi i toppen og bunden af h1-tag'et har padding sat til 6px! **50px - 6px - 6px = 38px**. Height og width-egenskaberne sætter højde og bredde på indholdet, uden padding, border og margin! Forklar det til Internet Explorer's box-model!

6. background-repeat: no-repeat;

Når man sætter **background-image**, er standard for hvordan det skal genskabe sig: **repeat**. De andre egenskaber hedder: **repeat-x**, **repeat-y** og **no-repeat**, hvilket af navn, burde være til at gennemskue.

Øvrige bemærkninger

Læg lige mærke til hvor mange tables der er brugt! ;)

Efter denne artikel, kunne du måske fristes til at tro at jeg ikke bryder mig meget om Internet Explorer. Jamen, hvor har du ret!! Jeg hader programmer som er fulde af fejl, ikke følger standarden og crasher i tide og utide!! Ville du ikke gøre det!?

Alt HTML og CSS kan du finde på <http://hermandsen.homepage.dk/exp/droptables/>.

Find evt. selv mere inspiration/information på <http://www.csszengarden.com/>.

//hermandsen

Kommentar af moejensen d. 21. Jan 2004 | 1

Kommentar af simonvalter d. 22. Jan 2004 | 2

ja helt sikkert et godt skrevet artikel, tror jeg sætter min lillebror til at læse den så han kan lave en hjemmeside til mig endag ;)

Kommentar af cms d. 23. Jan 2004 | 3

Pædagogisk, velskrevet og informativ :)

Kommentar af olebole d. 02. Apr 2006 | 4

En pletvis okay artikel, som dog lider af temmelig alvorlige mangler, der kan bidrage til at gøde alle de misforståelser, der flourer omkring DIV/CSS-design. Formålet med at opdele en side i data (XML),

udseende (CSS), funktionalitet (script) og markup (HTML/XHTML) er primært 'overskuelighed' - både for kodere og for browsere. Derfor har det heller aldrig været W3C's hensigt, at vi skulle kode med ID-selectors på stort set alle elementer ... tværtimod. Det site, der ofte bliver fremhævet csszengarden.com er et fint eksempel på, hvordan, det ikke bør gøres. CSS handler primært om nedarvning gennem klasser - og det synes jeg ikke fremgår med ønskelig tydelighed af denne artikel. Den kan endvidere bidrage til den totale misforståelse, at W3C er på vej bort fra tabeller. Tværtimod anbefales tabeller til kolonneopstillinger, som 80% af WWW's sider jo bruger. DIVs er gode til nogen ting og elendige til andre - tabeller er gode til andre ting og der er intet i vejen med at bruge tabeller, når de er bedst :)

Desuden synes det sært, at du disables browserens CSS-compatibilitet i det eneste HTML-eksempel, du skriver. Er det med vilje - eller grunder det mon i virkeligheden ikke i manglende CSS viden/forståelse? og når jeg taler om 'nedarvning gennem klasser', er det af hensyn til de 'tungeste i klassen' (som jeg åbenbart alligevel har tabt med den kommentar ;o). Pointen er, at bruger man et hav af ID-selectors, går meningen med CSS fløjten - da man så mister store dele af den overskuelighed, der netop er en af fordelene ved CSS

Kommentar af cyigen d. 23. Jan 2004 | 5

SUPER

Kommentar af erve d. 25. Jan 2004 | 6

Skide god artikel Jesper. God til hurtig indføring i emnet. Jeg er dog ikke blevet overbevist om tabeller uanvendelighed? Positionering via styles kræver jo osse en fedten rundt med pixels og %

Kommentar af jensgram d. 22. Jan 2004 | 7

Hvis dette ville blive standarden for artikler, så kunne forfatterne godt vælge mellem "50 points", "50 points" eller "50 points", når der skulle skrives :) Velformuleret, informativ og dybdegående...

Kommentar af el_barto_30_maj d. 21. Jan 2004 | 8

Rigtig god artikel, men bærer lidt præg af at du som du selv indrømmer er "farvet", altså besidder en vis modstand imod tabeller.

Jeg ser intet problem i at lave sider med tabeller, og så styre udseendet med CSS.

Ofte er det meget hurtigere end at skulle til at bestemme absolutte mål, højre/venstre justering osv.

Kommentar af mik789 d. 27. Jan 2004 | 9

God artikel. Men lige en ting: denne linje: `<?xml version="1.0" encoding="iso-8859-1"?>` (den såkaldte "xml prolog") skal udelades, hvis man vil have css til at virke fuldt ud i Internet Explorer (endnu en fejl ved denne browser). Der må IKKE stå noget som helst før doctype erklæringen, for så holder IE op med at overholde standarderne (går i "quirks mode"). Bare en detalje... men den kan godt drille

Kommentar af bufferzone d. 26. Jan 2004 | 10

Vigtig artikel af flere grunde. Dels får CSS en mere fremtrædende rolle i den nye standard, og så kan CSS bruges som optimering til søgemaskinerne ved at tune koden og få indholdet langt frem på siden. Artiklen er alle point værd

Kommentar af roenving d. 22. Jan 2004 | 11

El_barto har helt ret !-)

Kommentar af skovenborg d. 27. Jan 2004 | 12

En rigtig god artikel om et meget omtalt og diskuteret emne - rart at en eller anden tager det op. Der er selvfølgelig lige et par detaljer som fx brugen af position:absolute; (der ikke nødvendigvis betyder at den placerer elementet i forhold til siden. Fremgår ikke så tydeligt) og inherit som ikke understøttes af IE. Så kan man jo altid diskutere bag efter om tables er godt nok eller ej - denne artikel giver i hvert fald et rigtig godt eksempel på alternativet :-)

Kommentar af athlon-pascal d. 21. Jan 2004 | 13

Fed artikel, den ville jeg være villig til at betale 30 eller flere points for!

Kommentar af tofferman d. 13. Feb 2004 | 14

God artikel. Jeg er enig i at css og xhtml er vejen frem, også når vi taler om fremtidssikring i forhold til surfing på nettet ikke længere er forbeholdt en pc med en browser, men også skal bruges til mobiltelefoner, spillekonsoller, køleskabe og sikkert også brødristerne på et tidspunkt ;o)

Men man skal vænne sig til det. Jeg har selv rodet meget med det på toflo.dk, og det er ikke perfekt endnu. Men det giver en fleksibilitet og en simplificering som er guld værd.

Kommentar af fredie89 d. 10. Sep 2004 | 15

tables? Tabeller må du vel mene.. Nu har jeg ikke læst hele artiklen da jeg holder mig til tabeller. Allerede efter de første linier kan man tydelig se at du har misforstået noget. Hvorfor sætte tomme billeder ind når du kan bruge cellpadding/cellspacing? Jeg forstår udmærket godt at tabeller kan være lidt drilske men når man først har lavet det ser det godt ud og kan ses af en hver browser. Hvorfor skal man bruge utallige colspan/rowspan's? Alt i alt må man sige at du skal prøve tabellen og alle dens smarte funktioner :)

Kommentar af qtax87 (nedlagt brugerprofil) d. 17. Sep 2008 | 16

Jeg synes ikke er jeg blev meget klogere af denne artikel.

Til nogen ting er divs og spans gode hvor der er andre eksempler på hvor tables er bedre.

Jeg kunne måske personligt godt tænke mig en artikel eller mere beskrivelse af DOCTYPE osv.

Kommentar af stich d. 19. Jan 2006 | 17

Fin artikel!

Det er direkte forkert det olebole siger, med at CSS primært handler omkring nedarvning gennem klasser. 'Nedarvning' afleder man jo hurtigt ud fra akronymet, men at det hovedsageligt bør gøres via klasser er CSS1-tankegangen, den er forældet.

Og tabeller anbefales på ingen måde til kolonneopstillinger, tværtimod!

Det er kun IE der går i quirksmode pga. XML-deklarationen, men den skal jo ofte behandles særligt alligevel, når man vil lave avancerede ting med CSS...

Kommentar af jih d. 19. Feb 2004 | 18

helt sikkert den bedste artikel jeg har læst indtil videre...! så kan man komme af med de irriterende tables, og alle de tags der følger med dem! og layout'et skifter sådan set ik særligt meget... bliver jo ligefrem SMUKT! :o)

Kommentar af exp d. 21. Jan 2004 | 19

Helt igennem sk*degod artikel!
Nem at følge og godt skrevet.

Kommentar af lordhead d. 21. Jan 2004 | 20

En rigtig fed artikel! Jeg har selv lært noget nyt, selvom jeg kendte en del i forvejen - altid rart at blive klogere. Desuden skriver du det i et sprog der er til at forstå, og hader IE hvilket også er godt :) Du har også en anderledes men fed måde at linke til eksterne stylesheets, som er meget mere overskuelig end mine stylesheets. En rigtig fed artikel!!

Kommentar af mikze d. 11. Jun 2004 | 21

Fin artikel og til og blive klog af! Men jeg holder mig nu til tables :P

Kommentar af sandbox d. 22. Jan 2004 | 22

Tjah, de andre anmeldere har jo sagt det hele. Rigtig god artikel.

Kommentar af fotobob d. 08. Nov 2004 | 23

Kommentar af zeroaim d. 10. Feb 2004 | 24

En KANON GOD artikel !!!
Tusind tak for hjælpen

Mvh
Henrik

Kommentar af quit d. 26. Oct 2004 | 25

Kommentar af griffendk d. 13. Jan 2006 | 26

Kommentar af PeterFjelstrup d. 12. Jan 2011 | 27

Det givne eksempel opdeler jo siden i to spalter ligesom tabeller.
Jeg savner en måde at lægge en meny ind midt på siden, så den ligger ovenpå teksten, der så må krybe udenom - ligesom et billede på Word. Desuden en funktion, der lægger siden midt på en hvilken som helst opløsning.

Ellers lærte jeg meget. Og resten af diskussionen er jeg vel for 'ung' til at deltage i.